McNALLY

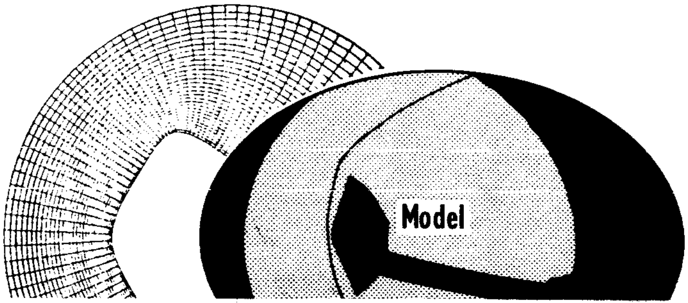
IN-64-TM

252677

276P

# Workshop on Numerical Grid Generation Techniques for Partial Differential Equations

## October 6-7, 1980



(NASA-TM-101828) WORKSHOP ON NUMERICAL TECHNIQUES FOR PARTIAL DIFFERENTIAL EQUATIONS. VOLUME 2: POSTER PAPERS (NASA) 276 p

N90-71381

Unclas

00/64 0252677

*Jointly Sponsored by Langley Research Center*

*and the*

*Institute for Computer Applications in Science and Engineering*

NASA
National Aeronautics and
Space Administration
Langley Research Center
Hampton, Virginia

Volume 2
Poster Papers

## POSTER SESSION I (Monday, October 6)

1. "Generation of Orthogonal Boundary-Fitted Coordinate Systems"
Roderick M. Coleman

2. "Fast Generation of Body Conforming Grids for 3-D Axial Turbomachinery Flow Calculations"
Djordje S. Dulikravich

3. "Boundary-Fitted Coordinates for Regions with Highly-Curved Boundaries and Reentrant Boundaries"
U. Ghia, K. N. Ghia

4. "A Two Dimensional Mesh Verification Algorithm"
Bruce Simpson

5. "Grid Generation for Two-Dimensional Finite Element Flowfield Computation"
Kenneth E. Tatum

6. "Conformal Mappings of Multiply-Connected Regions onto Regions with Specified Boundary Shapes"
Andrew N. Harrington

7. "Finite Difference Grid Generation by Multivariate Blending Function Interpolation"
P. G. Anderson and L. W. Spradley

8. "Component-Adaptive Grid Embedding"
E. Atta

9. "A Simple Numerical Orthogonal Coordinate Generator for Fluid Dynamic Applications"
Randolph A. Graves, Jr.

10. "A Three-Dimensional Body-Fitted Coordinate System for Flow Field Calculations on Asymmetric Nose Tips"
Darryl W. Hall

11. "Application of the Multigrid Method to Grid Generation"
Samuel Ohring

12. "Finite Difference Mesh Modification Technique with Application to Supersonic Wedge Flow"
Fred W. Martin
Paper not available for printing.

# Generation of Orthogonal Boundary-Fitted Coordinate Systems

Roderick M. Coleman

Computation, Mathematics, and Logistics Department
David W. Taylor Naval Ship Research and Development Center
Bethesda, Maryland 20084

# ABSTRACT

A method is presented for computing orthogonal boundary-fitted coordinate systems for geometries with coordinate distributions specified on all boundaries. The system which has found most extensive use in generating boundary-fitted grids is made up of the Poisson equations

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= P \\ \eta_{xx} + \eta_{yy} &= Q \end{aligned} \qquad (1)$$

The functions P and Q provide a means for controlling the spacing and density of grid lines in the coordinate system. Since all calculations are done in the computational plane, the dependent and independent variables in Equation (1) are interchanged, giving the usual transformed equations

$$\begin{aligned} \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_{\xi} + Qx_{\eta}) &= 0 \\ \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_{\xi} + Qy_{\eta}) &= 0 \end{aligned} \qquad (2)$$

where

$$\alpha = x_{\eta}^2 + y_{\eta}^2 \qquad \beta = x_{\xi}x_{\eta} + y_{\xi}y_{\eta}$$

$$\gamma = x_{\xi}^2 + y_{\xi}^2 \qquad J = x_{\xi}y_{\eta} - x_{\eta}y_{\xi}$$

The condition for orthogonality, i.e., $\xi$ = constant lines perpendicular to $\eta$ = constant lines, is $\beta = 0$, because

$$\beta = 0 \Rightarrow x_{\xi}/y_{\xi} = -y_{\eta}/x_{\eta}$$

which is equivalent to

$$1/y_x\Big|_{\eta = \text{constant}} = -y_x\Big|_{\xi = \text{constant}}$$

As a generating system based entirely on $\beta$, we consider

$$\beta_{\xi} = \beta_{\eta} = 0 \qquad (3)$$

which can have an orthogonal solution only when $\beta = 0$ at the corners of the computational region. An iterative solution of the generating system given in Equation (3) is applied successfully to several geometries. While questions remain concerning the existence and uniqueness of orthogonal systems, the generating method presented here adds to the available, useful techniques for constructing these systems.

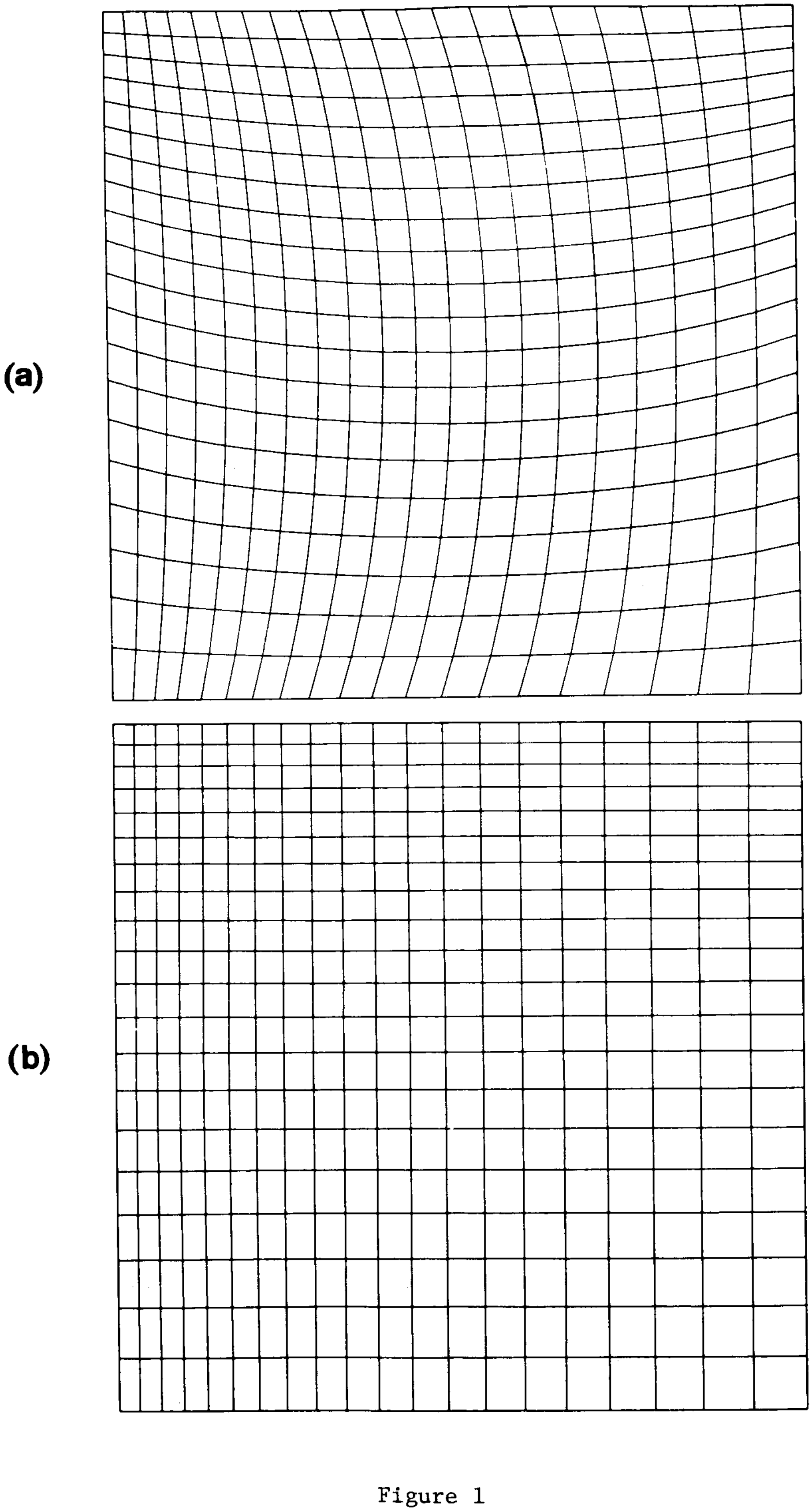


Figure 1

Figure 1 provides a comparison of two grids generated for a square region with nonuniform boundary coordinate spacing in both vertical and horizontal directions. The nonorthogonal mesh shown in Fig. 1a was generated using the Poisson system given by Equation (2) with $P \equiv Q \equiv 0$. Equation (2) was replaced with central difference formulae and the resulting system was solved by successive overrelaxation (SOR). The orthogonal mesh shown in Fig. 1b was obtained using Equation (3) as a generating system. Equation (3) was expanded and each derivative was replaced with the appropriate central difference formula. Again, the resulting system was solved by SOR iteration.

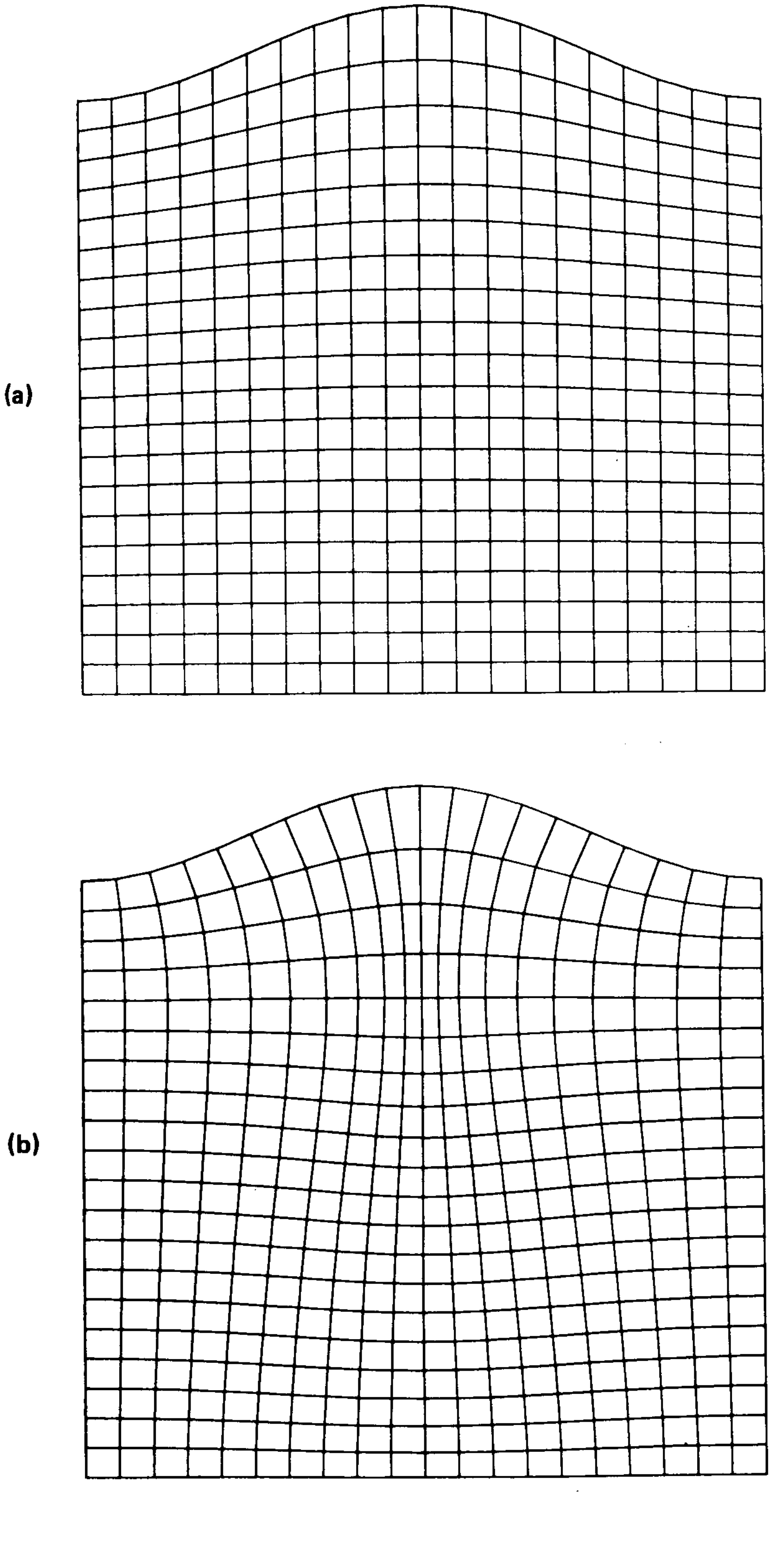


Figure 2

Two 21 x 21 girds generated for a simply-connected region with one convex boundary are shown in Figure 2. Fig. 2a shows a nonorthogonal coordinate system generated by Equation (2) with $P \equiv Q \equiv 0$ (a Laplace system); Fig. 2b shows a coordinate system generated by Equation (3). Note the orthogonality of the coordinate lines intersecting the curved upper boundary in Fig. 2 and the resultant bending of these lines in the interior.

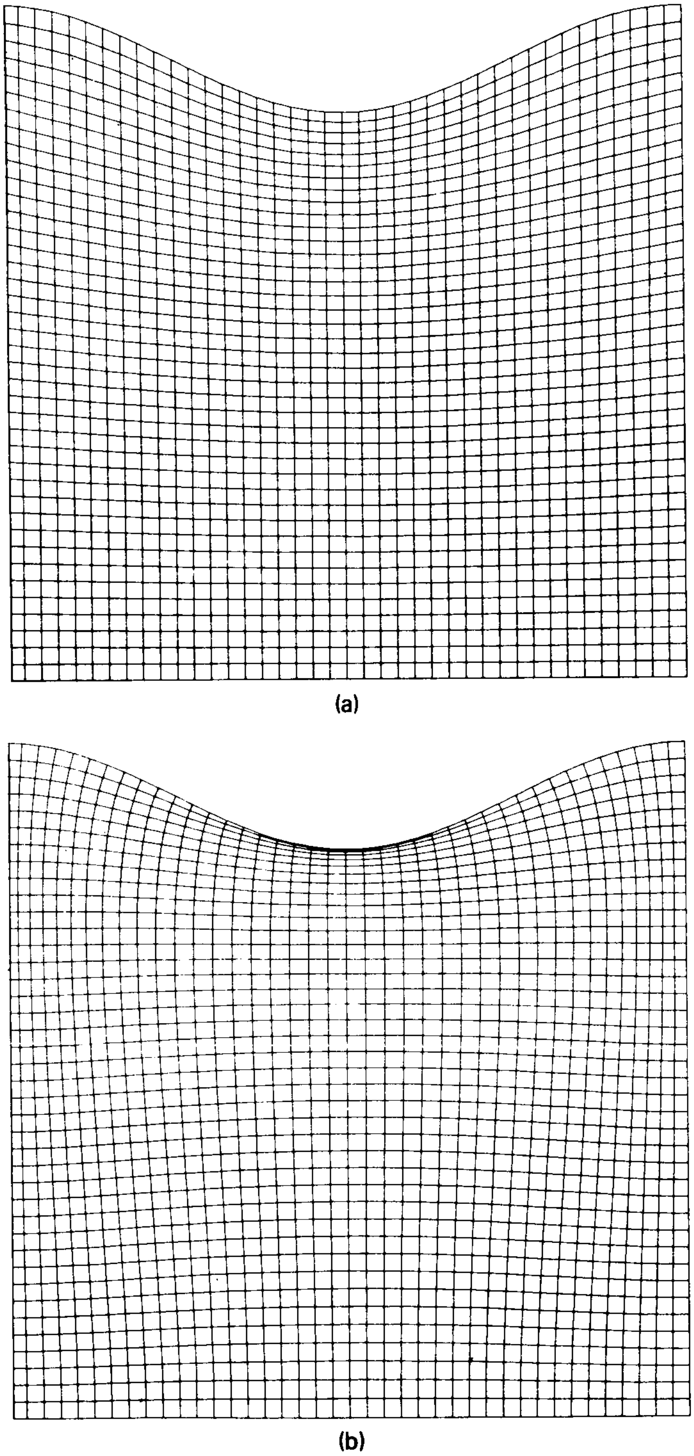


Figure 3

Figure 3 shows a region similar to that of Fig. 2 with a concave rather than convex curved boundary. As before, Fig. 3a shows a Laplace-generated grid and Fig. 3b shows a β-generated grid obtained using Equation (3). The orthogonal mesh must have rather fine spacing near the concave upper boundary to accommodate the curvature. To verify that the fine mesh spacing in Fig. 3b is due to the geometry and not to a singularity in the transformation, we have refined the mesh as seen in the next figure.

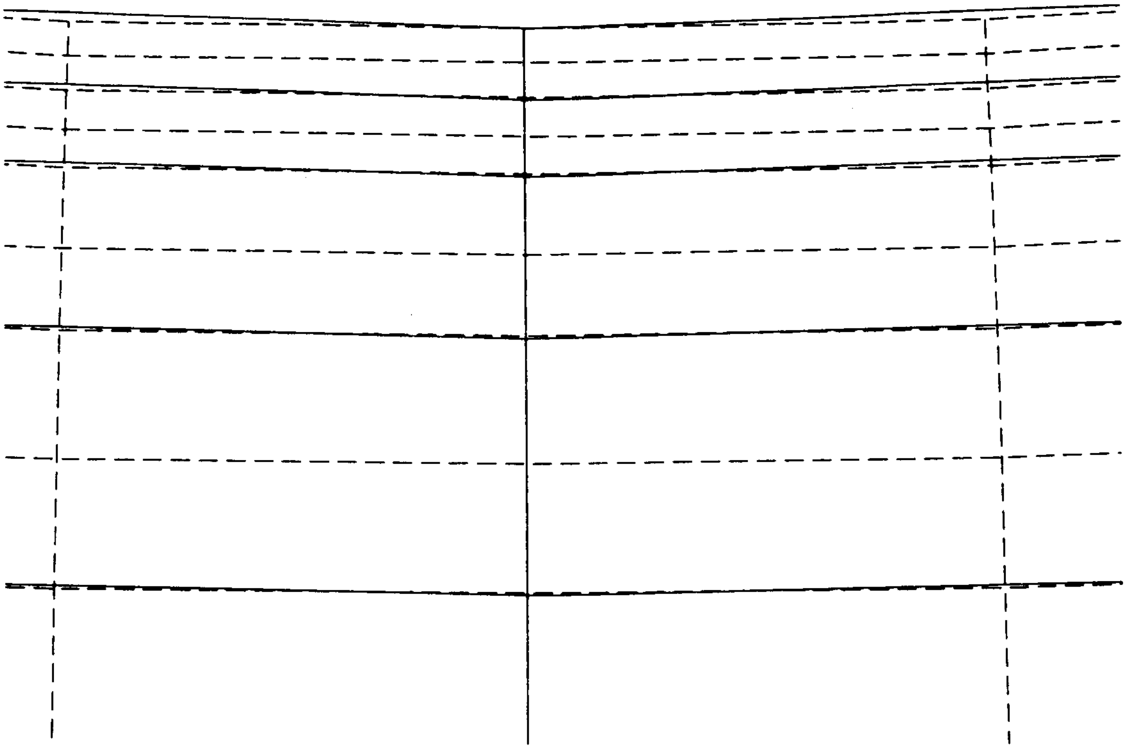
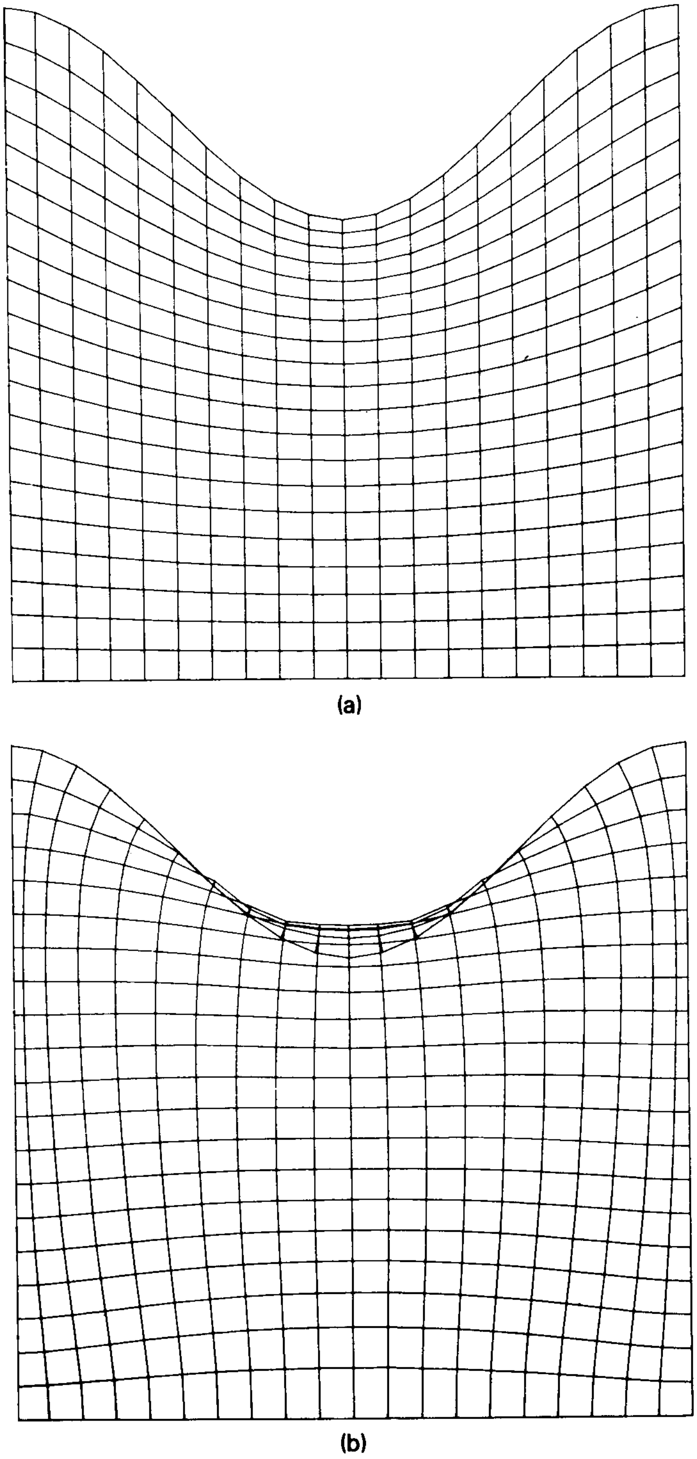


Figure 4

Figure 4 compares two different grids, one coarse with 1681 points and the other fine with 6561 points, generated for the concave region. The fact that corresponding grid lines are in about the same position in both meshes confirms that the coarse discretization yields a good approximate solution to the exact problem. A further confirmation comes from consideration of the Jacobian at the midpoint of the upper boundary. The value of the Jacobian computed on the coarse mesh is nonzero and agrees very well with the value computed on the fine mesh. There is no indication of a zero Jacobian anywhere in the region.



(a)

(b)

Figure 5

To demonstrate some of the problems that can arise, we attempted to generate an orthogonal mesh on a region similar to the previous one but with greater curvature of the concave boundary. The grid shown in Fig. 5a was generated by a Laplace system and the unacceptable grid in Fig. 5b was generated by the system of Equation (3). To verify that a mesh with crossing lines can also be produced by a Poisson system, we computed directly the forcing functions P and Q using Equation (2) with x and y as given in Fig. 5b. We then solved Equation (2) iteratively for x and y using this P and Q, and regenerated the grid of Fig. 5b.

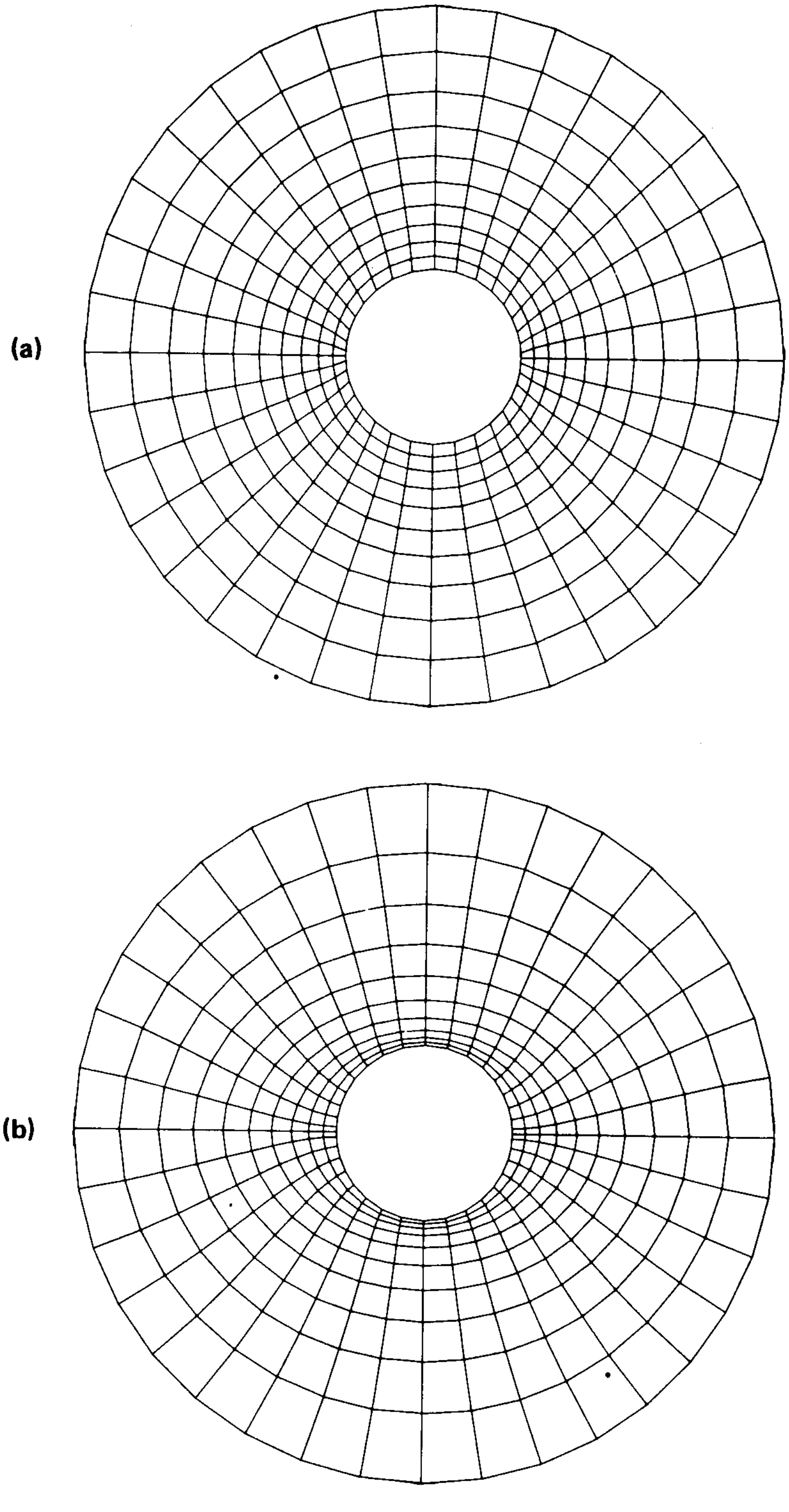


Figure 6

As the final example, we considered a doubly-connected region bounded by concentric circles as shown in Fig. 6. Since this region is symmetric with respect to any line passing through the center, each grid was generated for half the region and reflected in the line of symmetry. The symmetry line was treated as a boundary with fixed coordinate distribution, thus assuring that $\beta = 0$ at the corners of the computational region. The spacing on the outer boundary, but not on the inner boundary, was uniform. Had the spacing on both boundaries been uniform, the grid produced by the Laplace generating system (Fig. 6a) would have been the usual polar coordinate system which is orthogonal. In Figs. 6a and 6b, the line of symmetry was taken as a horizontal line through the center of the figure. The mesh of 6a was used as an initial guess for the iterative procedure used to obtain the mesh of 6b.

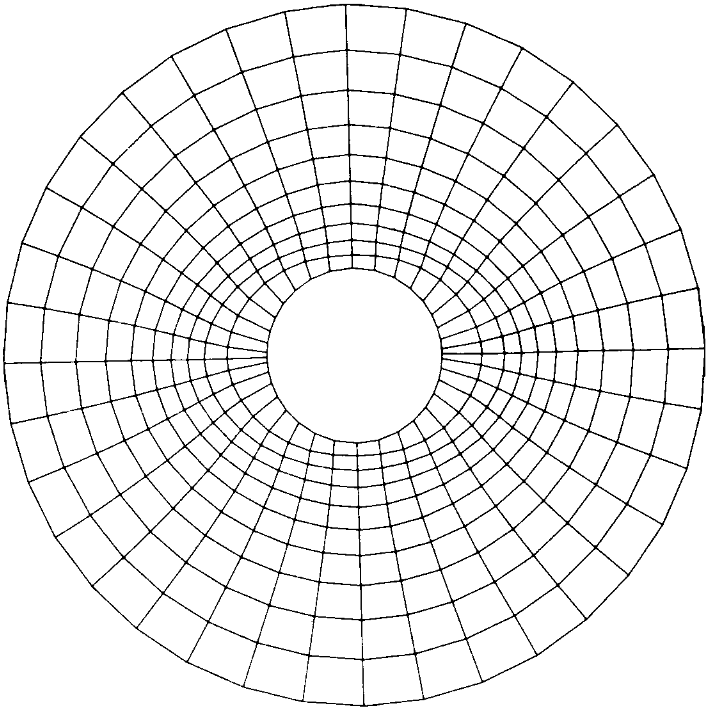


Figure 7

In Fig. 7, we show a β-generated grid computed for the same doubly-connected region used in the previous figure. As before, the mesh of Fig. 6a was used for the initial guess, but in this case the line of symmetry was taken as a vertical line through the center. Interestingly, the two orthogonal grids generated for the same physical region (Figs. 6b and 7) are quite dissimilar because different points were held constant after the same initial guess.

# FAST GENERATION OF BODY CONFORMING GRIDS FOR 3-D AXIAL TURBOMACHINERY FLOW CALCULATIONS

by Djordje S. Dulikravich

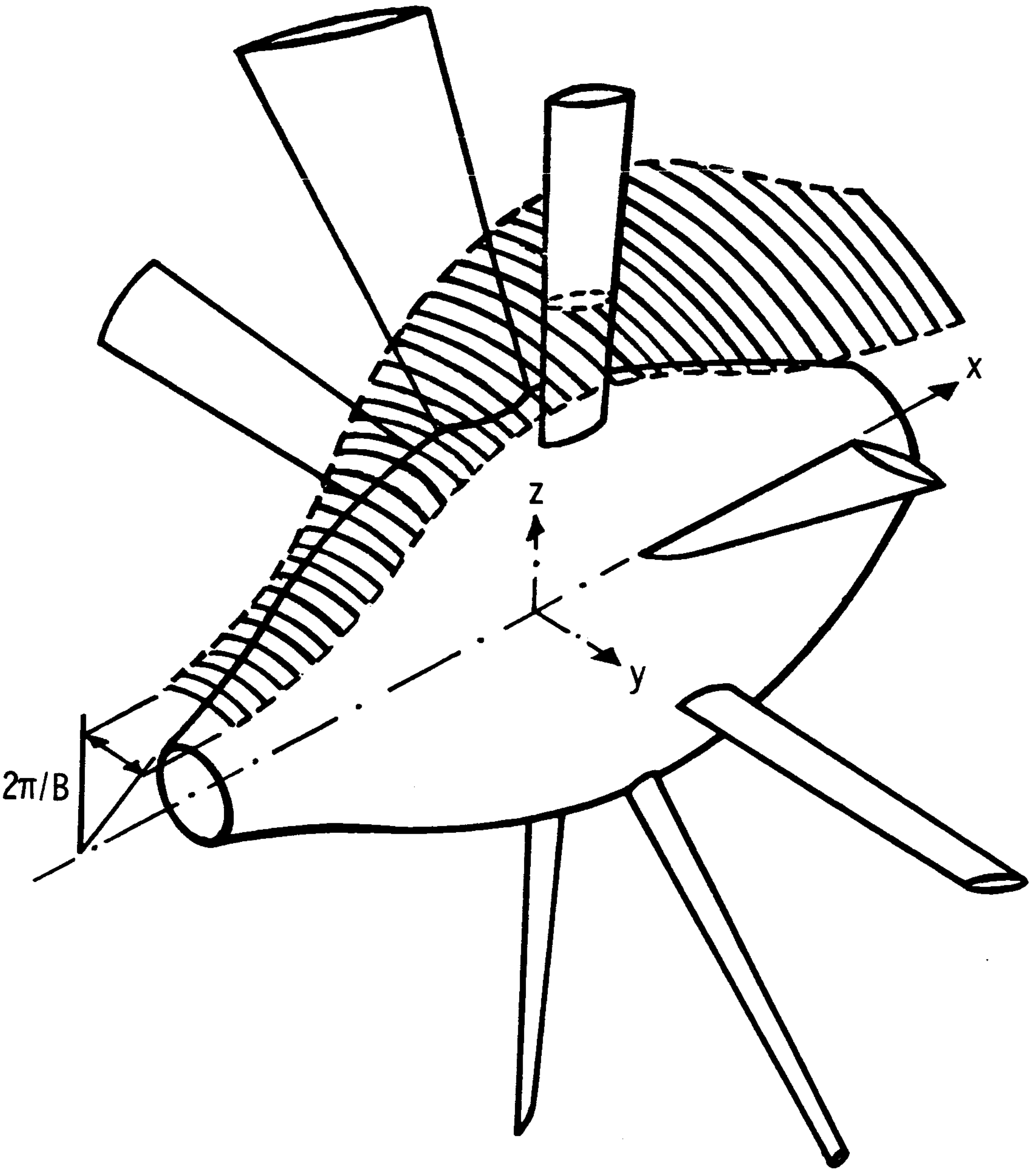
# ABSTRACT

A fast algorithm has been developed for accurately generating boundary conforming, three-dimensional, consequtively refined, computational grids applicable to arbitrary axial turbomachinery geometry. The method is based on using a single analytic function to generate two-dimensional grids on a number of coaxial axisymmetric surfaces positioned between the hub and the shroud. These grids are of the "O"-type and are characterized by quasi-orthogonality, geometric periodicity and an adequate resolution throughout the flowfield. Due to the built-in additional nonorthogonal coordinate stretching and shearing, the grid lines leaving the trailing edge of the blade end at downstream infinity, thus simplifying the numerical treatment of the three-dimensional trailing vortex sheet.

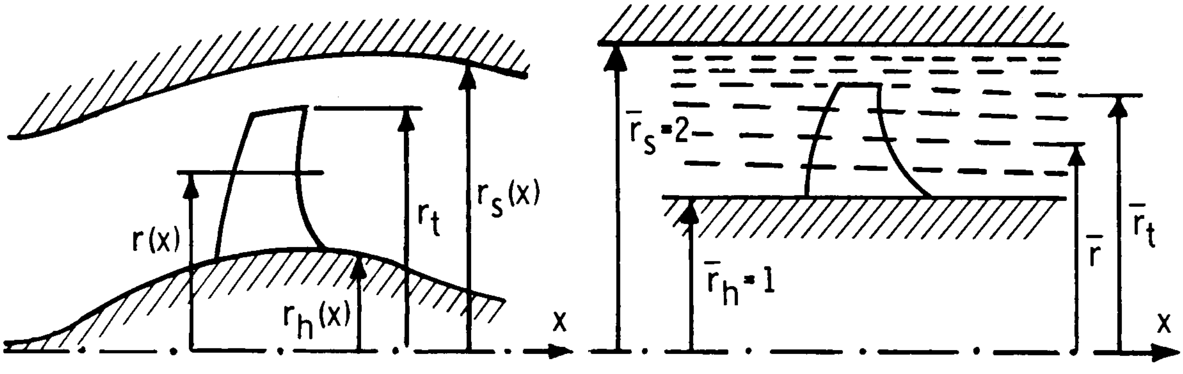
*NASA - National Research Council Research Associate, NASA-Lewis Research Center, Computational Fluid Mechanics Branch, Cleveland, Ohio 44135. Presently a visiting research scientist at DFVLR-Göttingen Universität, F.R. Germany.

The main objective of this work is to develop a fast algorithm for generating body-conforming three-dimensional computational grids. An equally important objective is to preserve the high accuracy of the discretized representation of the solid boundaries. When analyzing steady flows through turbomachinery rotors and stators, it is sufficient to consider a single rotationally periodic segment of the flowfield. This segment is a doubly infinite strip stretching in the direction of the axis of rotation. The strip has a constant angular width of $2\pi/B$ where $B$ is the total number of blades. Each blade has an arbitrary spanwise distribution of taper, sweep, dihedral and twist angle. The local airfoil shapes can vary in an arbitrary fashion along the blade span. The rotor hub and the duct (or shroud) can have different arbitrary axisymmetric shapes.

Such an arbitrary three-dimensional physical domain (Fig. 1) is first discretized in the spanwise direction by a number of coaxial axisymmetric surfaces which are irregularly spaced between hub and shroud.

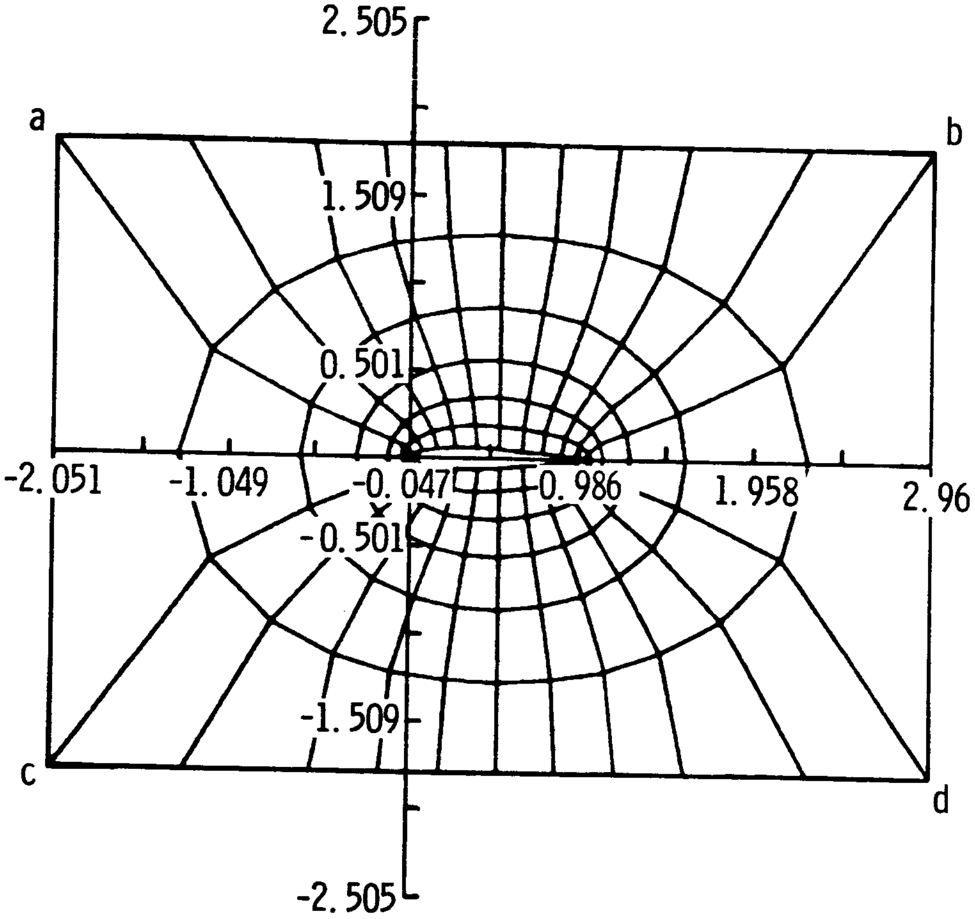


The major problem in generating the spanwise suriaces is an accurate determination of the intersection contours between the irregular blade surface and the coaxial axisymmetric surfaces cutting the blade. The coordinates of the points on these contours are defined by fitting cubic splines along the blade and interpolating at the radial stations corresponding to each axisymmetric surface $\bar{r}$ = const.

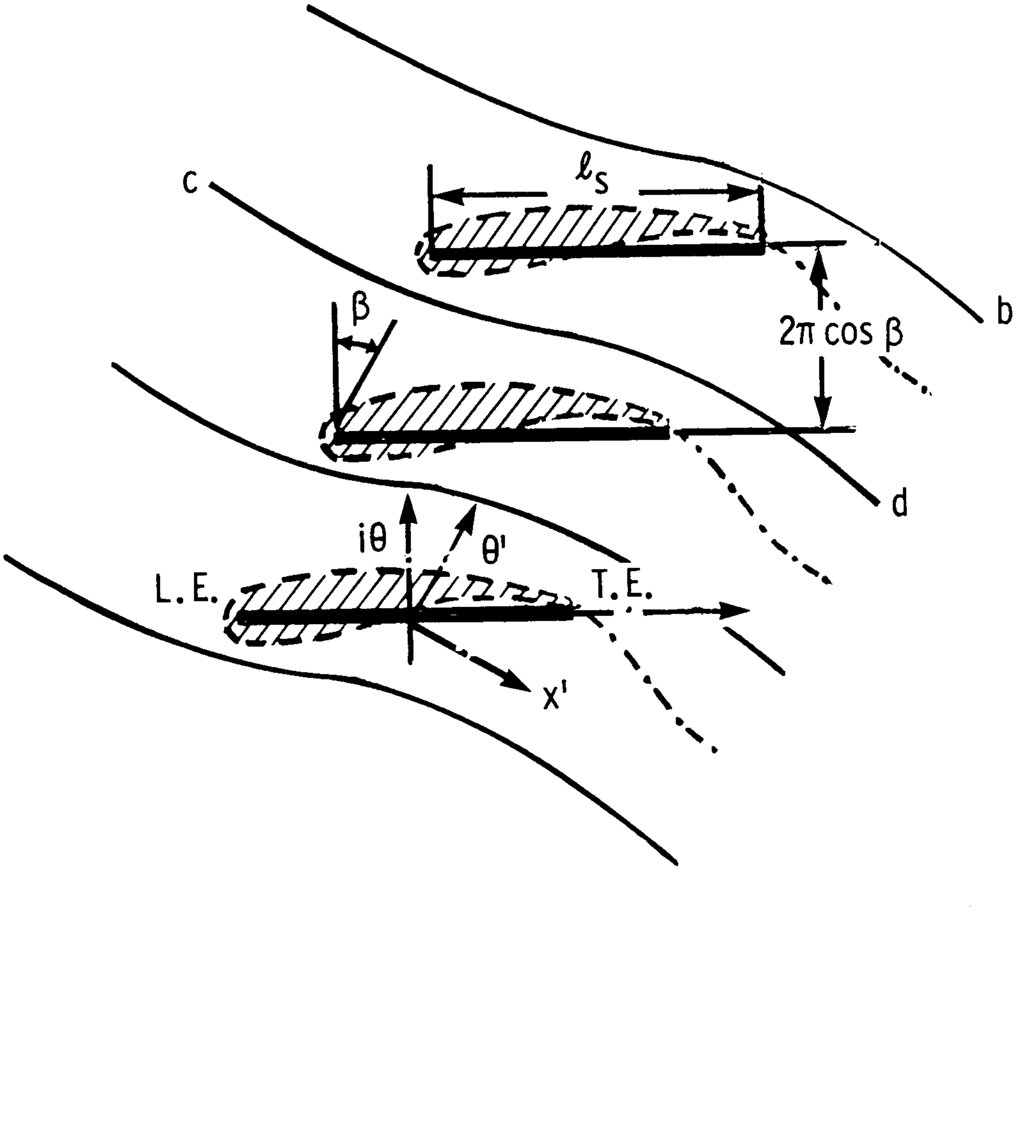


$$\bar{r} = 1 + \frac{r(x) - r_h(x)}{r_s(x) - r_h(x)}$$

The two-dimensional grid should have the following features: (a) grid cells should conform with the contour shape and the shape of the periodic boundaries $\overline{ab}$ and $\overline{cd}$, (b) grid should be geometrically periodic in the $\theta$'-direction meaning that the grid points along the periodic boundary $\overline{ab}$ must have the same respective x'-coordinates as the grid points along the periodic boundary $\overline{cd}$, (c) grid lines should not be excessively non-orthogonal in the vicinity of solid boundaries, (d) a grid line emanating from the trailing edge should end at downsteam infinity and (e) grid cells should be concentrated in the regions of high flow gradients.



Once the shape of the intersection contour on a particular cutting axisymmetric surface is known, the problem becomes one of discretizing a doubly-connected two-dimensional domain $\underset{\sim}{w} = x + i\theta$.



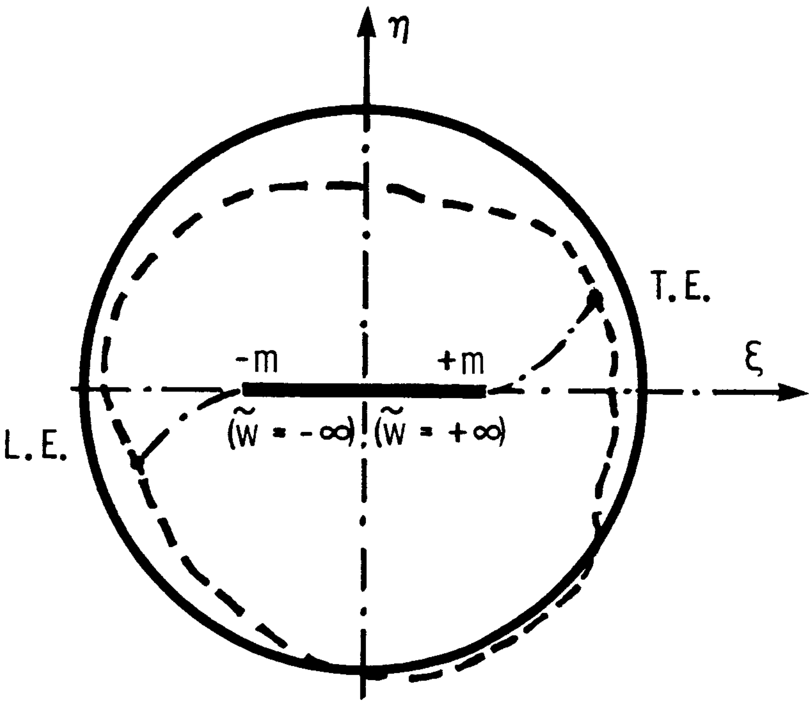
A grid with these properties can be most easily generated with the use of a single analytic function. One such function is

$$\tilde{w} = e^{i\beta} \ln\left(\frac{m - \tilde{z}}{m + \tilde{z}}\right) + e^{-i\beta} \ln\left(\frac{1 - m\tilde{z}}{1 + m\tilde{z}}\right); \; 0 < m < 1$$

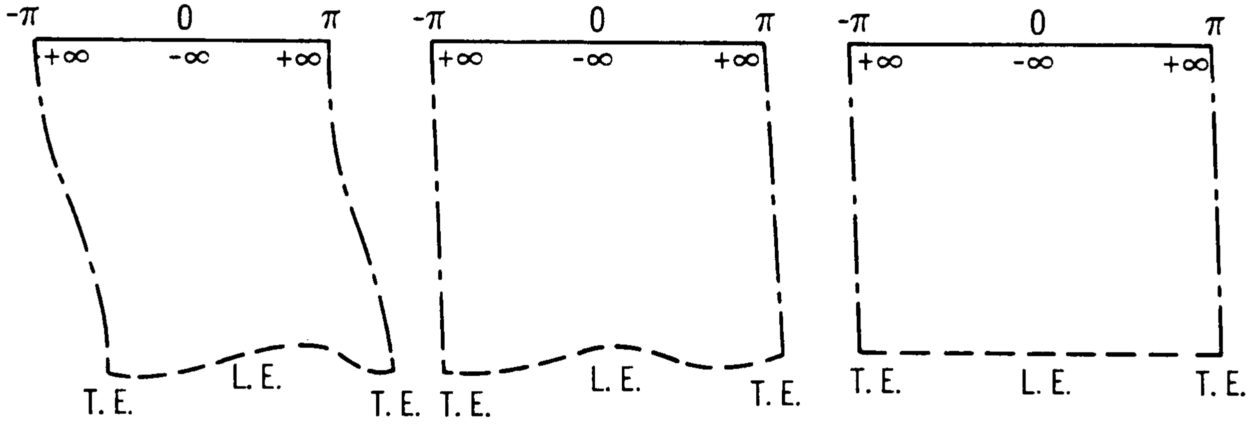
where $\tilde{w} = x + i\theta$ and $z = \xi + i\eta$. This complex function maps conformally a unit circle with a slit in the middle whose end-points are situated at $\tilde{z} = \pm m$ onto the cascade of straight slits in the $\tilde{\omega}$-plane. Each slit has a length $\ell_s$ where

$$\ell_s = 4\left(\cos\beta \sinh^{-1} \frac{2m \cos\beta}{1 - m^2} + \sin\beta \sin^{-1} \frac{2m \sin\beta}{1 + m^2}\right)$$

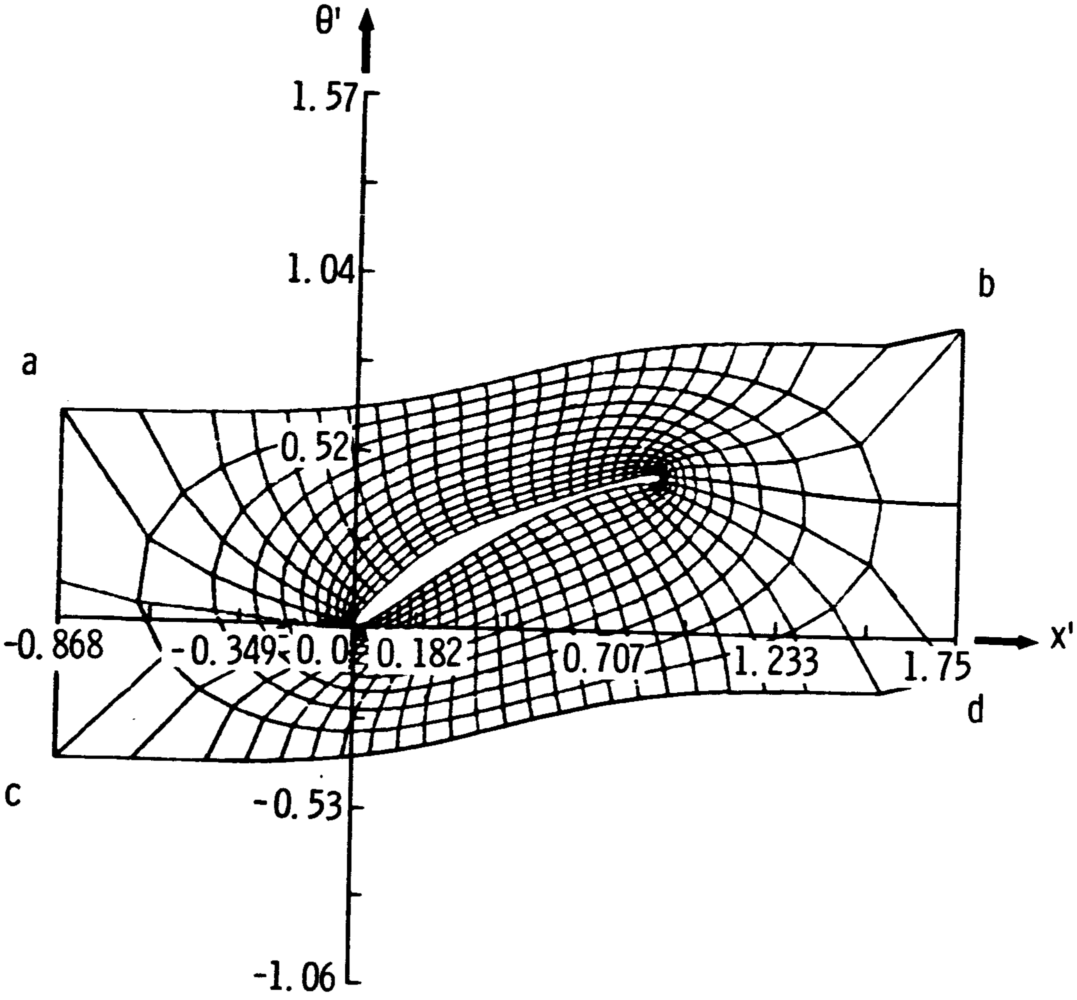
The slits are spaced $2\pi \cos\beta$ distance apart, where $\beta$ is the stagger angle of the cascade of slits.



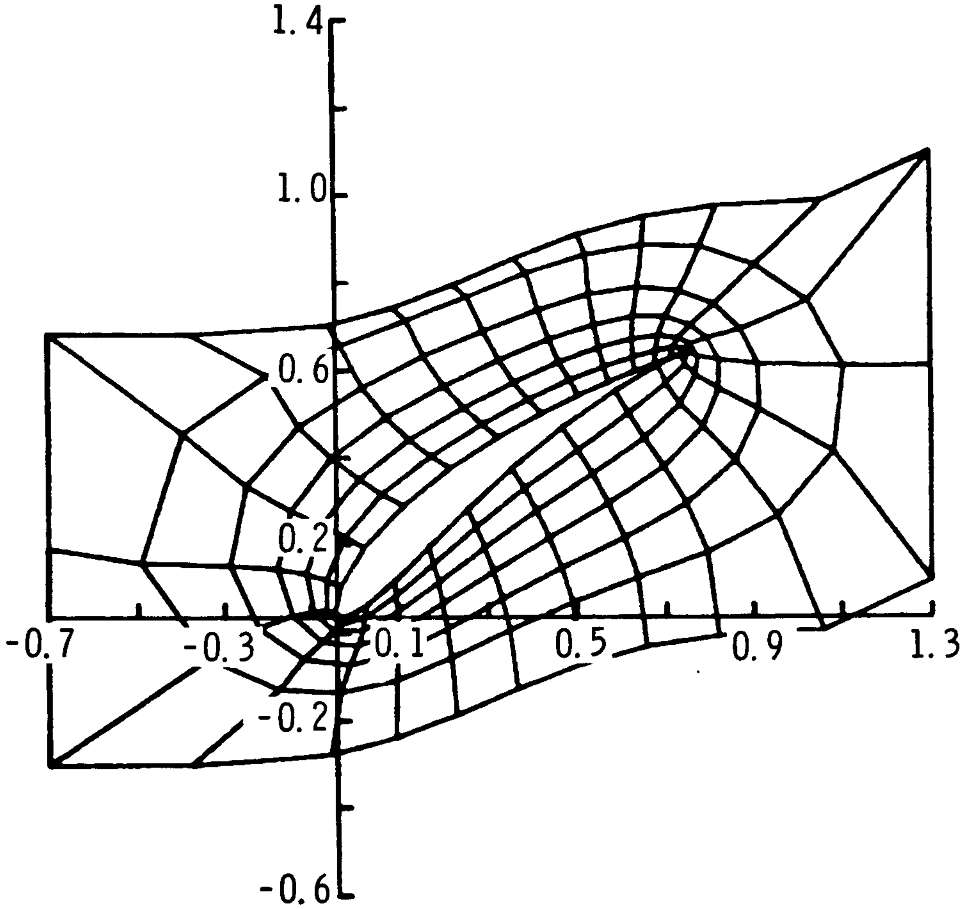
The unit circle is "unwrapped" using elliptic polar coordinates (2,4) resulting in a deformed rhomboidal shape which is then sheared in the horizontal and vertical direction (2) resulting in a rectangular (X,Y) computational domain.



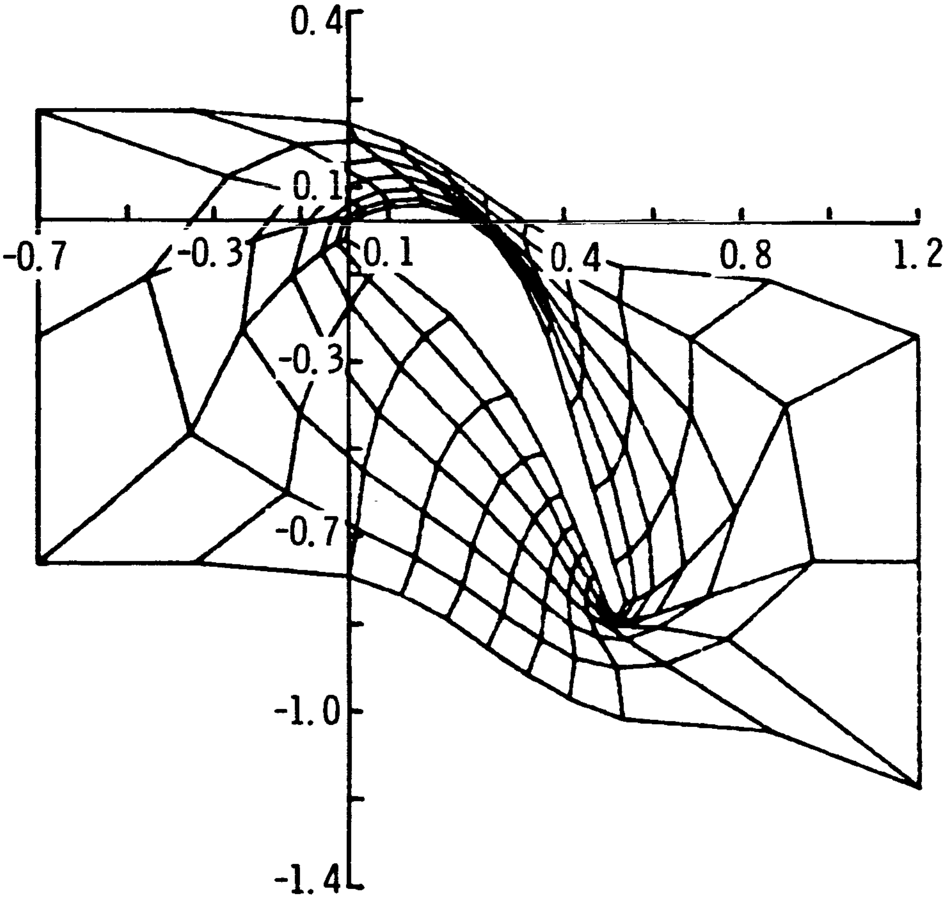
The transformation of an actual cascade of airfoils will result in a cascade of unit circles which are even more deformed. Consequently, more nonorthogonality will be introduced in the transformation by additional shearing of coordinates. A uniform grid in the (X,Y) plane which is symmetrically spaced with respect to the Y-axis, remaps back into the physical $(x,\theta)$ plane as an "O"-type boundary conforming grid. The actual radial coordinates are obtained by fitting cubic splines along the elliptic mesh lines and interpolating at a number of axial stations at which the radius of the corresponding axisymmetric surface is known.



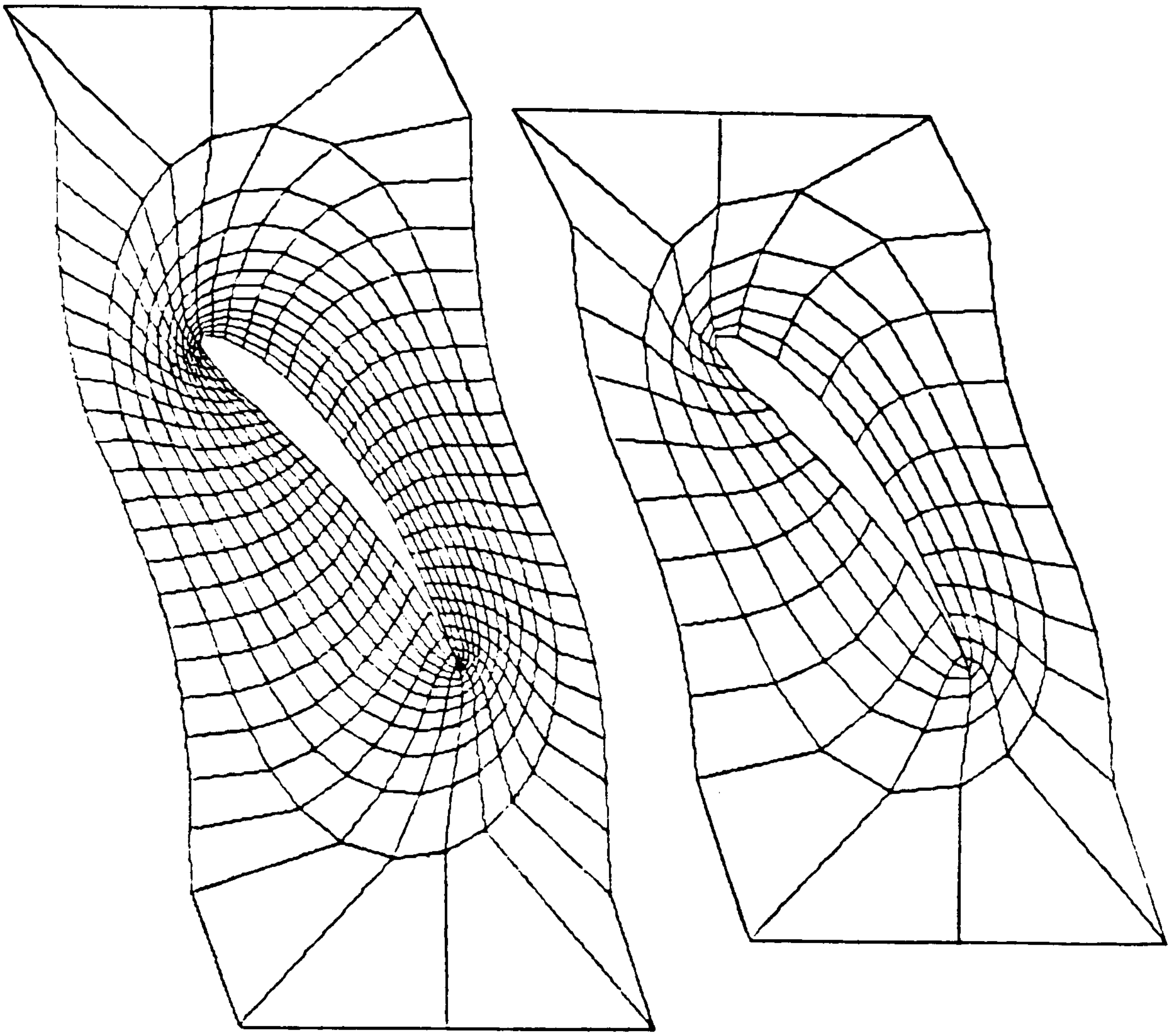
The present method is equally applicable to the blades with blunt (or rounded), wedged and cusped trailing and/or leading edges.



A disadvantage of the present method is that it is not applicable for the very thick, highly stagered blades which are very closely spaced. This problem can be resolved by using a different form of the mapping function; for example, one which mapps a cascade of circles into a cascade of circular arcs instead of a cascade of straight slots.



A sample run shows that it takes 10 seconds of CPU time on an IBM 3033 to generate (x, y, z) coordinates of two 3-D grids and to write them on two separate disks (5). The first (coarse) grid consisted of 27x9x8 points and the second (refined) grid has 51x15x14 poirts.



# REFERENCES

1. Dulikravich, D. S.; and Caughe,', D.:' Finite Volume Caluclation of Transonic Potential Flow Through Rotors and Fans. To be published in AIAA Journal, 1980.

2. Dulikravich, D. S.: Numerical Calculation of Inviscid Transonic Flow Through Rotors and Fans. Ph.D. Thesis, Cornell Univ., 1979.

3. Kober, H.: Dictionary of Conformal Representations, Dover Publications, Inc., Second ed. with corrections, 1957, p. 131.

4. Spiegel, M. R.: Mathematical Handbook of Formulas and Tables. Schaum's Outline Series, McGraw-Hill Book Co., 1968, p. 127.

5. Dulikravich, D. S.: "CAS3D-Fortran Program for 3D, Steady, Transonic, Potential, Axial Turbomachinery Flows." NASA TP, 1980.

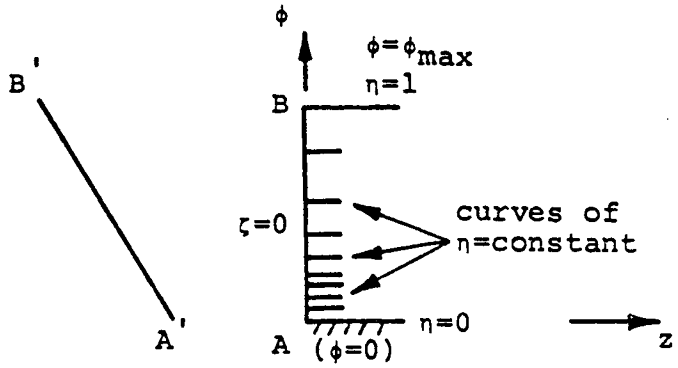
# BOUNDARY-FITTED COORDINATES FOR REGIONS WITH HIGHLY CURVED BOUNDARIES AND REENTRANT BOUNDARIES

U. GHIA AND K.N. GHIA

UNIVERSITY OF CINCINNATI, CINCINNATI, OHIO

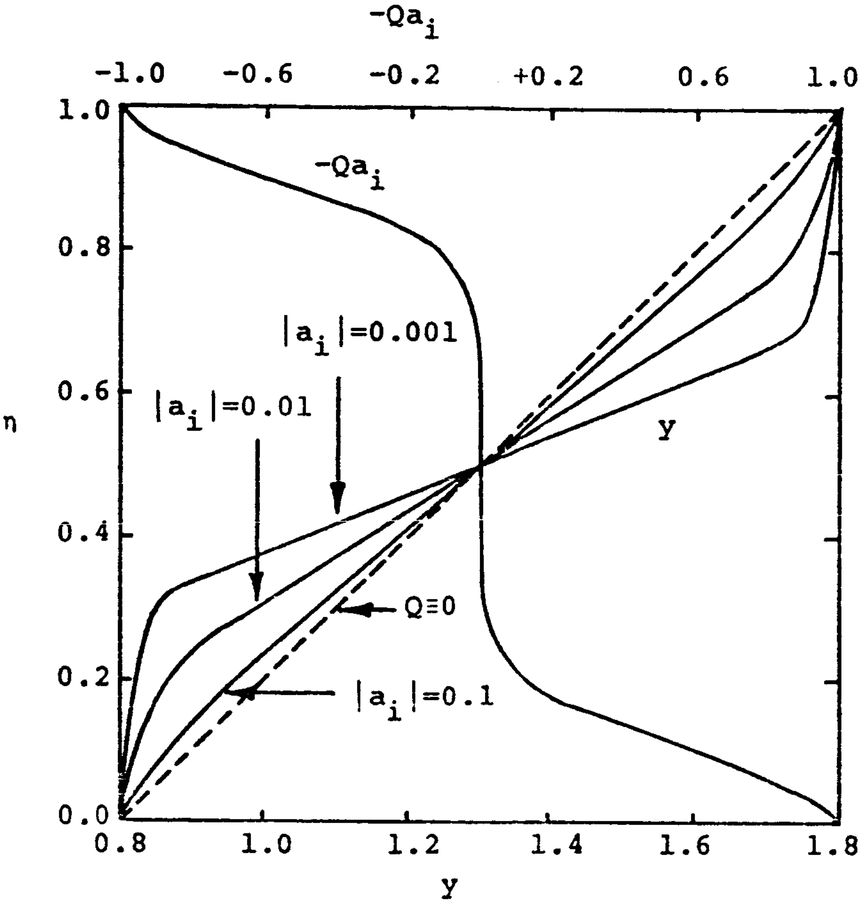
A procedure has been developed, using the differential-equation approach, for generating boundary-fitted coordinates for regions with highly curved boundaries as well as reentrant boundaries, such as those encountered in breaking surface waves. The resulting coordinates are nearly orthogonal and can provide adequate resolution even in the reentrant region. Consistent treatment of end boundaries and the use of a systematic initialization scheme and advanced implicit numerical solution techniques make the procedure highly efficient. The method developed for implicit enforcement of the periodicity boundary condition should be beneficial in the analysis of turbomachinery flow applications.

# CONSISTENT TREATMENT OF END-BOUNDARIES



A limiting form of the coordinate equations at the end-boundary is solved to determine, prior to the complete solution, the point distribution at this boundary, consistent with the interior distribution. This procedure avoids discontinuities in the transformed-coordinate derivatives near the end-boundaries, while maintaining Dirichlet boundary conditions for the transformation.

## SOLUTION OF LIMITING EQUATION AT END-BOUNDARY



$$\phi_{\eta\eta} + Q\,\phi_\eta^3 = 0$$

where

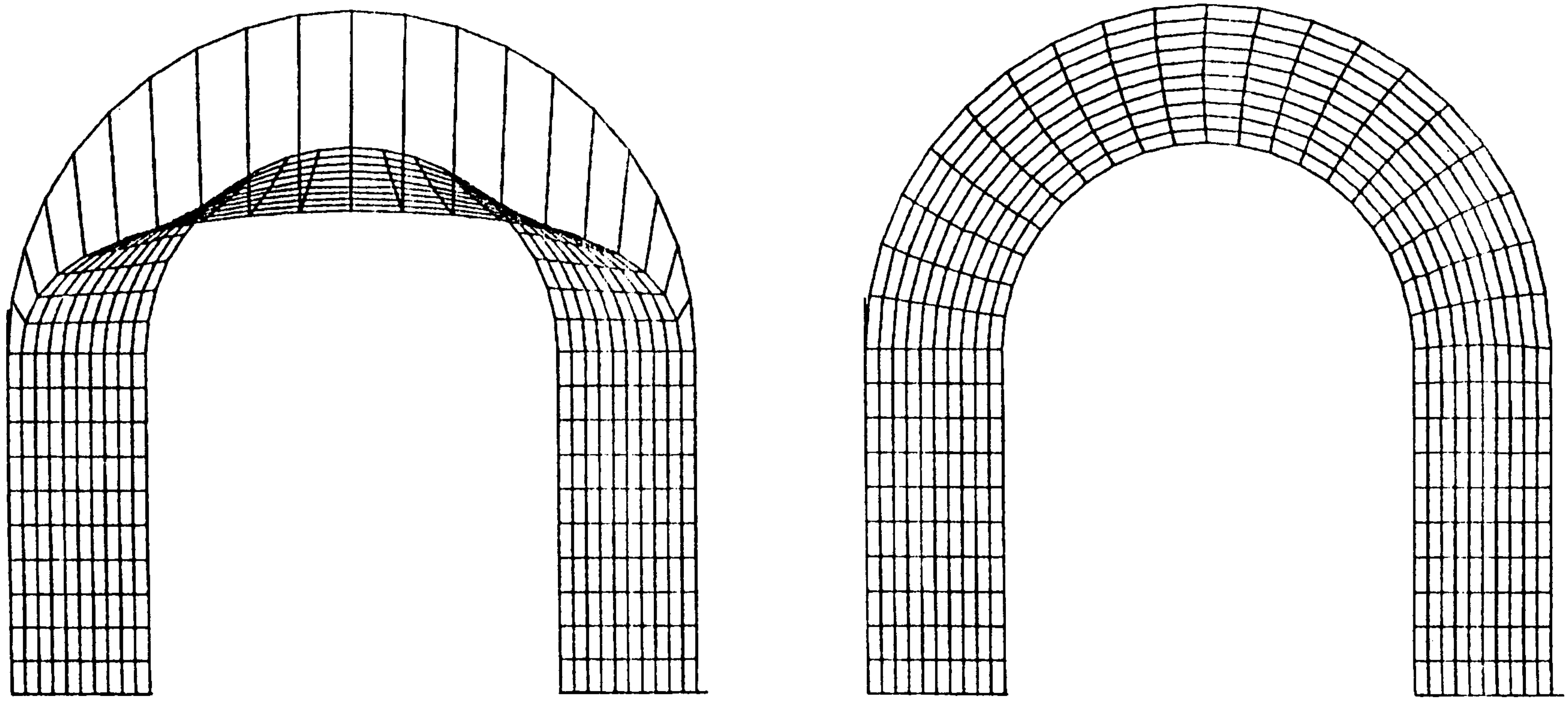
$$Q(\eta) = \sum_{k=1}^{2} \frac{1}{a_k} \exp[-(\eta-\eta_k)^2/(2b_k^2)]$$

$a_1 < 0$ , $|a_1| = a_2$

$b_1 = b_2 = 0.1$

$\eta_1 = 0$ , $\eta_2 = 1$

INITIALIZATION PROCEDURE

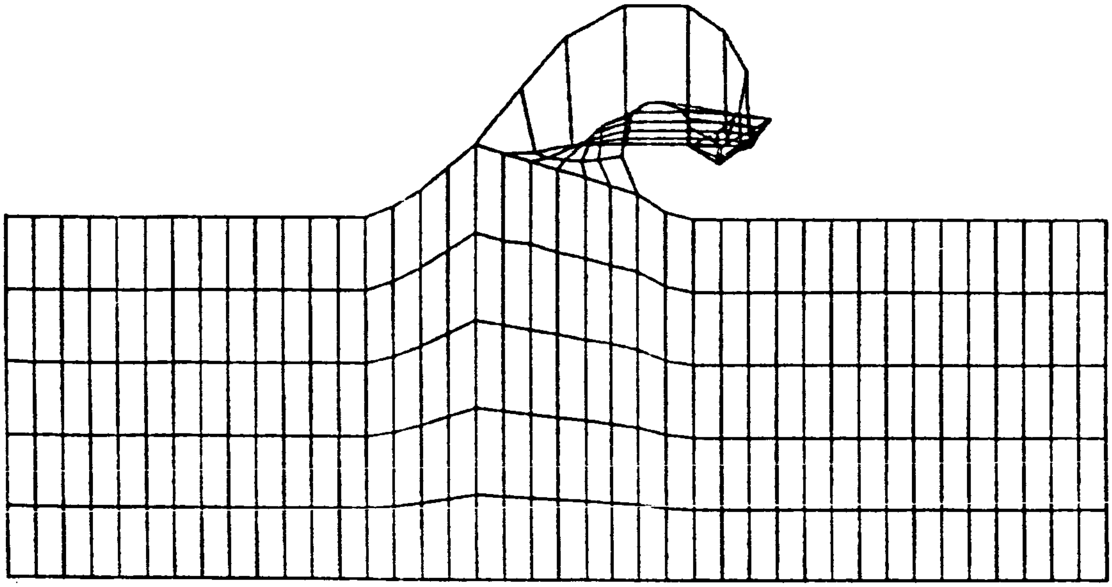


GEOMETRIC INITIALIZATION

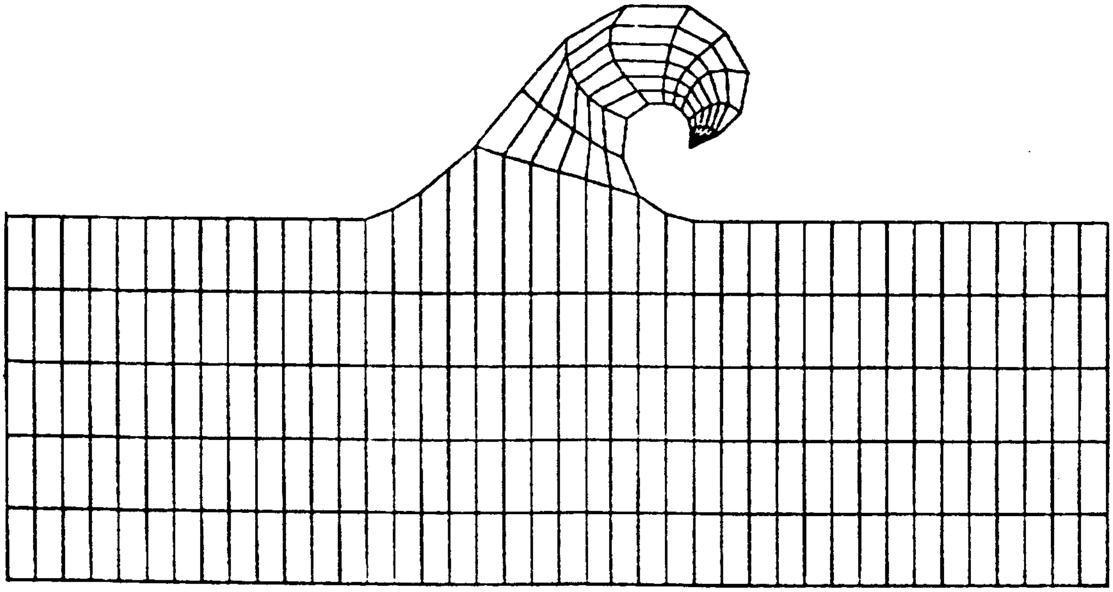
INITIALIZATION BY
LOCALLY SELF-SIMILAR SOLUTION

# INITIALIZATION PROCEDURE

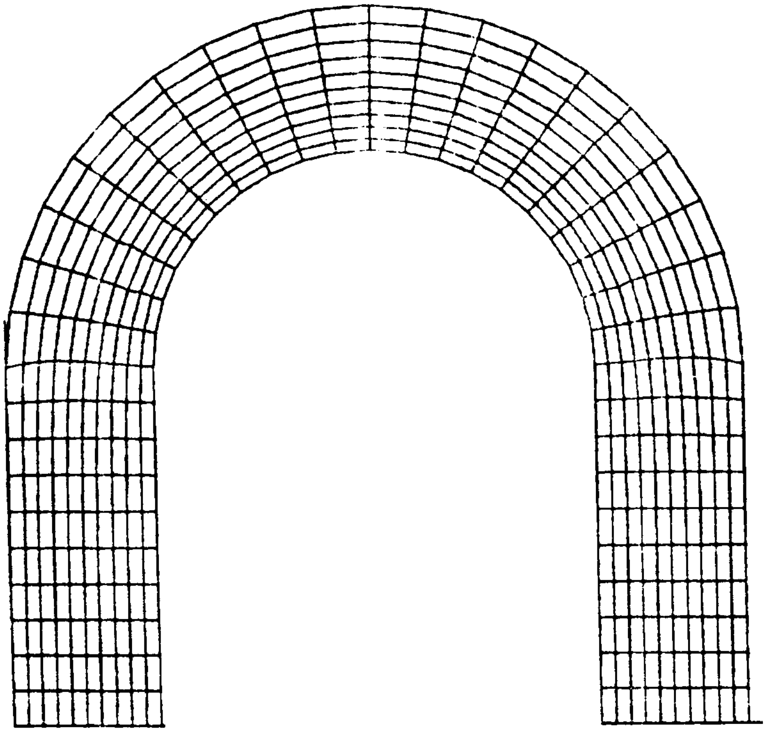
## GEOMETRIC INITIALIZATION



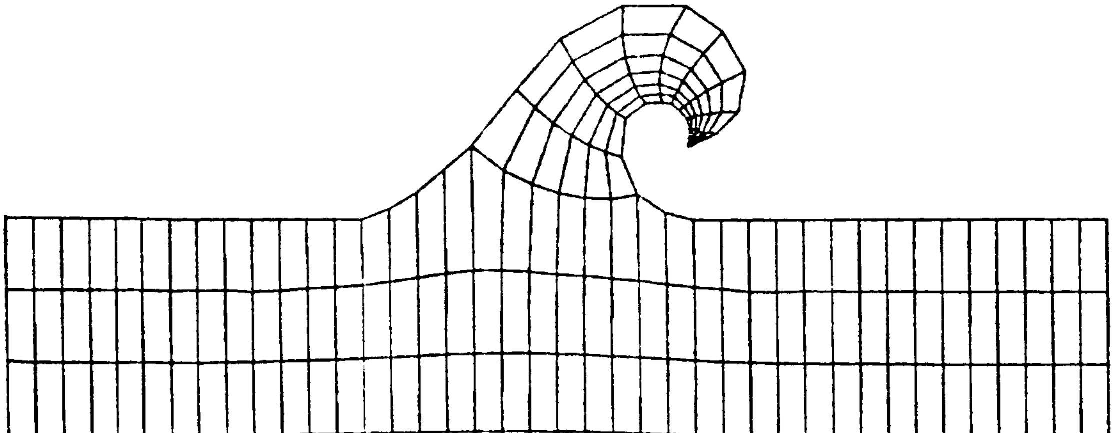
## INITIALIZATION BY LOCALLY SELF-SIMILAR SOLUTION



SURFACE-ORIENTED COORDINATES FOR DUCT WITH HIGHLY CURVED BOUNDARIES



BOUNDARY-ORIENTED COORDINATES FOR A TYPICAL SURFACE WAVE WITH REENTRANT BOUNDARIES



## IMPLICIT ENFORCEMENT OF PERIODICITY BOUNDARY CONDITION

### DIFFERENTIAL EQUATION:

$$\phi'' + a\phi = b$$

### PERIODICITY BOUNDARY CONDITIONS:

$$\phi_0 = \phi_1 = A \;; \qquad \phi_0' = \phi_1' = B$$

where A and B are unknown.

### SOLUTION PROCEDURE:

Let $\phi = Af + Bg + h$

$$f'' + af = 0 \qquad g'' + ag = 0 \qquad h'' + ah = b$$

with

$$f_0 = 1 \qquad g_0 = 0 \qquad h_0 = 0$$

$$f_1' = 0 \qquad g_1' = 1 \qquad h_1' = 0$$

Then,

$$Af_1 + Bg_1 + h_1 = A$$

$$Af_0' + Bg_0' + h_0' = B$$

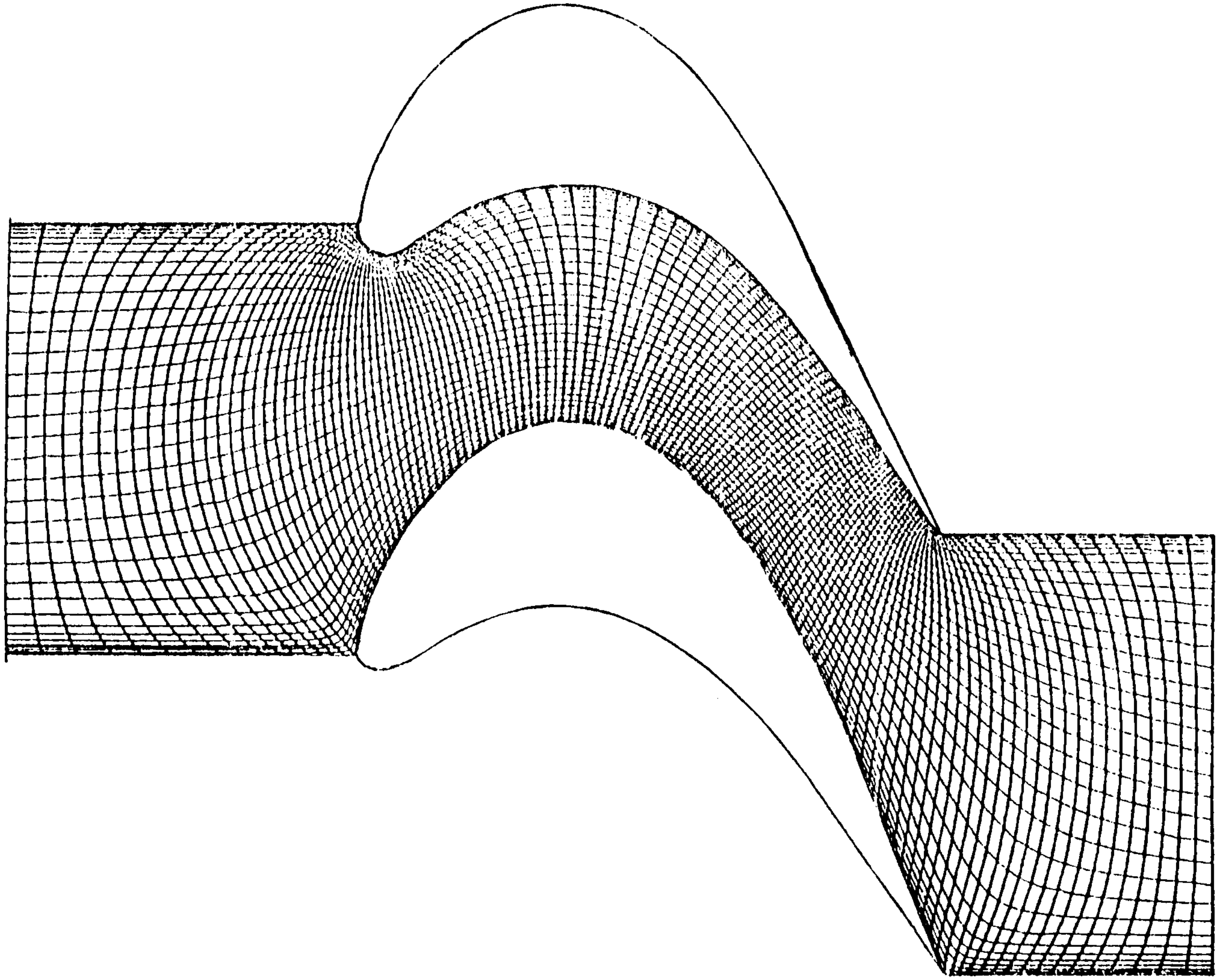
so that

$$A = [h_1(1-g_0') + h_0' \, g_1]/[(1-f_1)(1-g_0') - f_0' \, g_1]$$

$$B = [h_0'(1-f_1) + f_0' \, h_1]/[(1-f_1)(1-g_0') - f_0' \, g_1]$$

SURFACE-OREINTED COORDINATES FOR A TURBINE CASCADE -

(129 x 33) NONUNIFORM GRID WITH EASILY APPLICABLE PERIODICITY



STREAMWISE-ALIGNED SURFACE-ORIENTED COORDINATES FOR A TYPICAL TURBINE CASCADE - (161 x 33) NONUNIFORM GRID

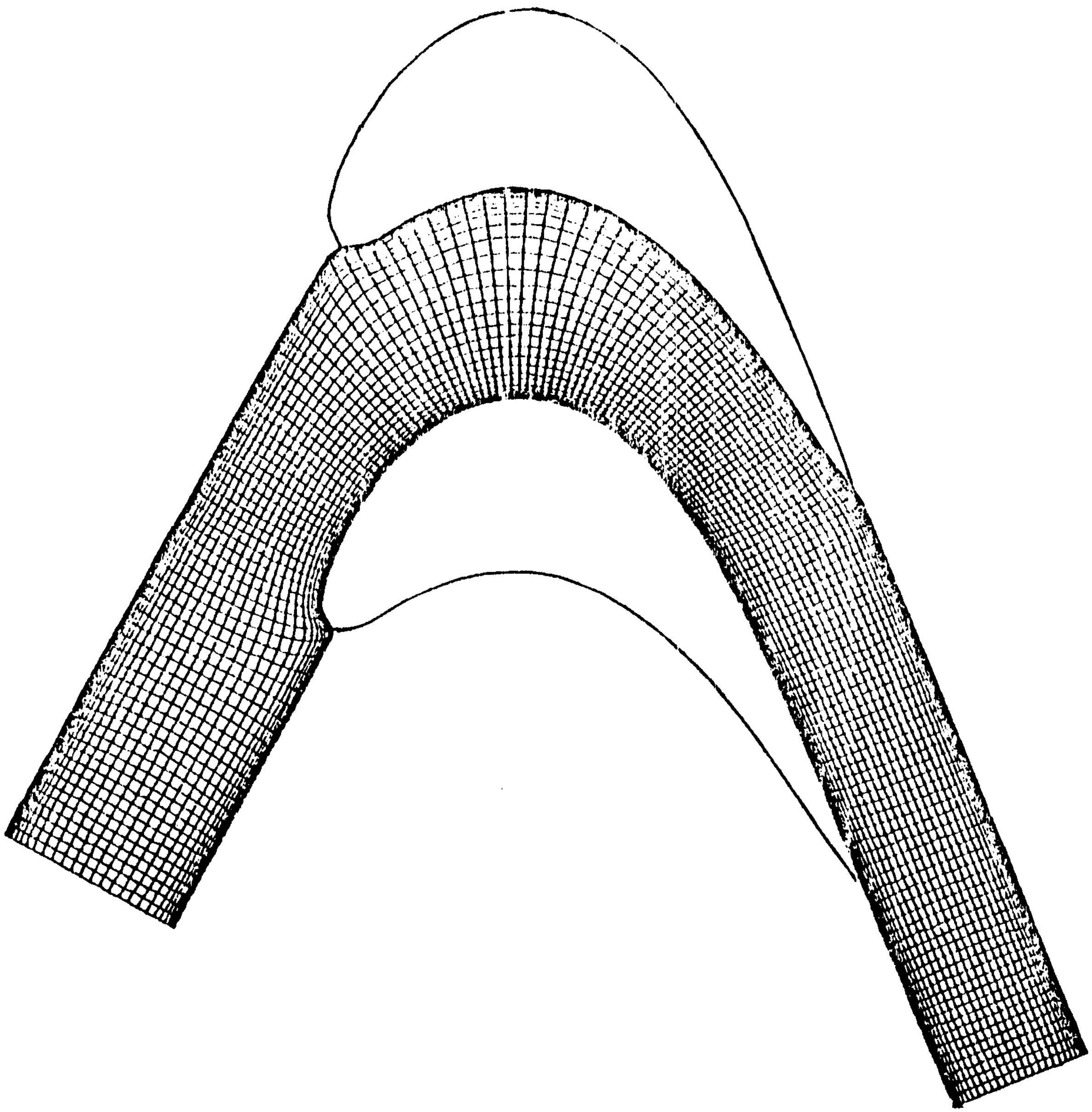


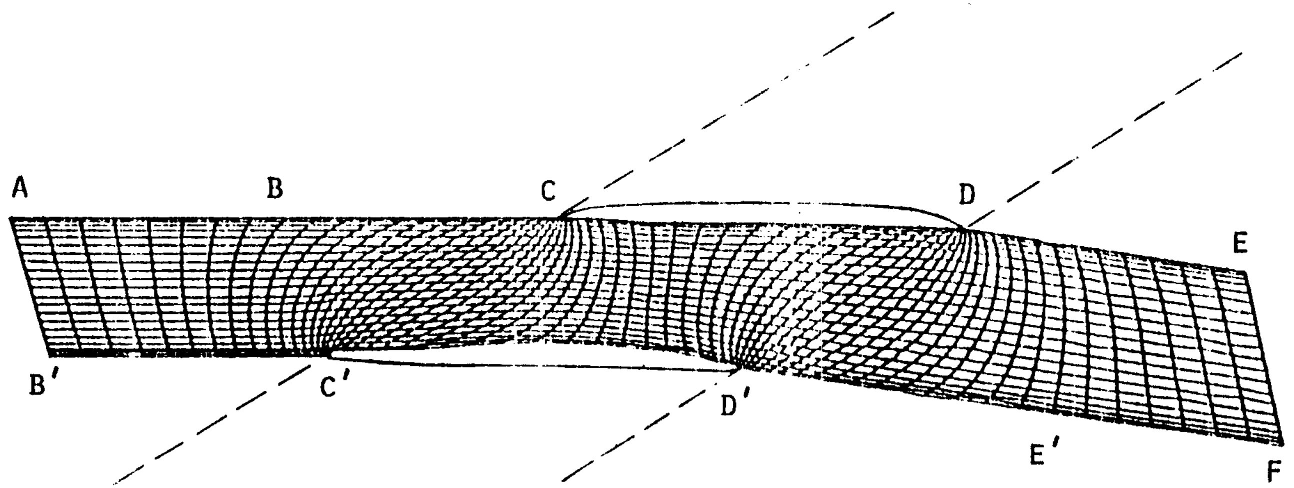
TABLE 1. EFFECT OF MULTIGRID (MG) ITERATION TECHNIQUE ON CONVERGENCE OF COORDINATE SOLUTION FOR CASCADE WITH EASILY APPLICABLE PERIODICITY

| Method | Grid | Work Units of Resp. Finest Grid | CPU Seconds | Remarks |
|---|---|---|---|---|
| ADI | (65 x 17) | 100 | 37.69 | uniform spacing |
| SIP | (65 x 17) | 53 | 11.96 | uniform.spacing |
| MG-SIP | (65 x 17) | 6.5 | 2.08 | uniform spacing |
| ADI | (65 x 17) | 95 | 36.67 | nonuniform spacing |
| SIP | (65 x 17) | 25 | 6.33 | nonuniform spacing |
| MG-SIP | (65 x 17) | 7.5 | 2.32 | nonuniform spacing |
| MG-SIP | (129 x 33) | 6.4 | 8.44 | nonuniform spacing |

TABLE 2. CONVERGENCE OF COORDINATE SOLUTION FOR CASCADE GEOMETRY WITH PERIODICITY USING A STRONGLY IMPLICIT PROCEDURE (SIP) AND MULTIGRID (MG) TECHNIQUE

| Method | Grid | Work Units of Resp. Finest Grid | CPU Seconds | Remarks |
|---|---|---|---|---|
| SIP | (161 x 33) | 81.00 | ≃100.0 | uniform spacing. convergence is one order less than for nonuniform spacing. |
| MG-SIP | (161 x 33) | 7.48 | 10.79 | uniform spacing |
| MG-SIP | (161 x 33) | 8.23 | 11.49 | nonuniform spacing |
| MG-SIP | (81 x 17) | 8.88 | 4.02 | nonuniform spacing |

## TYPICAL SURFACE-ORIENTED COORDINATES FOR A CASCADE WITH HIGH STAGGER - (65 x 21) NONUNIFORM GRID



This figure shows a multiple-circular-arc supersonic compressor cascade with a large stagger angle and a typical coordinate distribution for such a cascade. The grid lines are concentrated near the surface of both the blades, especially near their leading and trailing edges, in order to provide good resolution for the viscous and shock effects in these regions. In addition to the non-uniform distribution of the grid points, an effort has been made to maintain near-orthogonality wherever possible. The existing non-orthogonality can be easily removed by increasing the number of points in the streamwise direction, although the coordinate distribution shown in this figure may actually be preferred for supersonic cascades. Moreover, the point distribution along the free boundaries is such as to enable enforcement of the periodicity condition, i.e., the point distributions along BC and DE are the same as along B'C' and D'E', respectively. The number of working units required to generate the (65 x 21) coordinates shown was 8.44 using the SIP-multigrid method; the corresponding CPU time was 3.48 seconds.

## CONCLUSIONS

- Generation of coordinates for regions with highly curved boundaries requires suitable initial conditions; locally self-similar equations provide an excellent non-iterative initial solution.

- Generation of appropriate Dirichlet boundary conditions even with non-zero forcing functions enhances solution convergence rate.

- Use of implicit numerical solution procedures together with the multigrid iteration technique constitutes an effective method for solution of the nonlinear governing differential equations with large number of grid points.

- An adaptive coordinate distribution is formulated for the breaking surface-wave problem with a reentrant boundary; solutions are presently being obtained for a free surface wave starting from an initial sinusoidal form and undergoing the breaking phenomenon.

# A Two Dimensional Mesh Verification Algorithm

R. Bruce Simpson
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada.

## Abstract

A finite element mesh is commonly represented in a program by lists of data i.e. vertex coordinates, element incidences, boundary data. In general, these lists describe a collection of triangles. Whether the triangles form proper mesh for a region or not, i.e. whether they 'tile' a region, is data dependent in a non obvious way. This paper specifies a set of conditions on the triangles (i.e. on the list data) which ensure that the triangles tile a region and which also can be verified by an algorithm which is referred to in the title, and which is claimed to be of reasonable efficiency.

## Basic List Representation of a Mesh

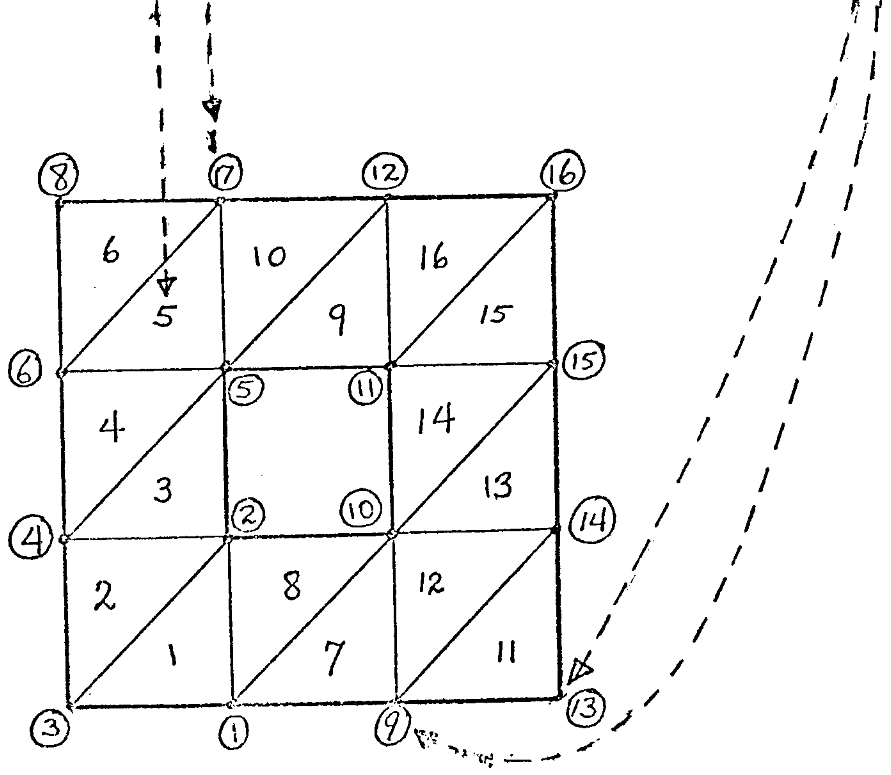
The mesh verification algorithm assumes that the collection of triangles is described by three lists as shown in the following small example.

| Vertex Index | Coordinates x-y coordinates | |
|---|---|---|
| 1 | 2.00 | 1.00 |
| 2 | 2.00 | 2.00 |
| 3 | 1.00 | 1.00 |
| 4 | 1.00 | 2.00 |
| 5 | 2.00 | 3.00 |
| 6 | 1.00 | 3.00 |
| 7 | 2.00 | 4.00 |
| 8 | 1.00 | 4.00 |
| 9 | 3.00 | 1.00 |
| 10 | 3.00 | 2.00 |
| 11 | 3.00 | 3.00 |
| 12 | 3.00 | 4.00 |
| 13 | 4.00 | 1.00 |
| 14 | 4.00 | 2.00 |
| 15 | 4.00 | 3.00 |
| 16 | 4.00 | 4.00 |

| Element incidences index | vertex indices | | |
|---|---|---|---|
| 1 | 1 | 2 | 3 |
| 2 | 2 | 4 | 3 |
| 3 | 2 | 5 | 4 |
| 4 | 5 | 6 | 4 |
| 5 | 5 | 7 | 6 |
| 6 | 7 | 8 | 6 |
| 7 | 9 | 10 | 1 |
| 8 | 10 | 2 | 1 |
| 9 | 11 | 12 | 5 |
| 10 | 12 | 7 | 5 |
| 11 | 13 | 14 | 9 |
| 12 | 14 | 10 | 9 |
| 13 | 14 | 15 | 10 |
| 14 | 15 | 11 | 10 |
| 15 | 15 | 16 | 11 |
| 16 | 16 | 12 | 11 |

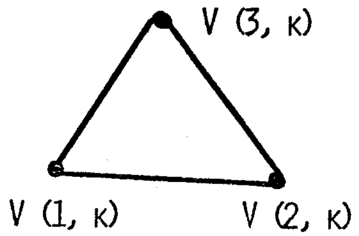
| Boundary References index | references | |
|---|---|---|
| 1 | 1 | 3 |
| 2 | 2 | 2 |
| 3 | 3 | 1 |
| 4 | 4 | 2 |
| 5 | 6 | 1 |
| 6 | 6 | 2 |
| 7 | 7 | 3 |
| 8 | 8 | 1 |
| 9 | 9 | 3 |
| 10 | 10 | 1 |
| 11 | 11 | 1 |
| 12 | 11 | 3 |
| 13 | 13 | 1 |
| 14 | 14 | 2 |
| 15 | 15 | 1 |
| 16 | 16 | 1 |

indicates a boundary edge starts at —▷ – – ▷



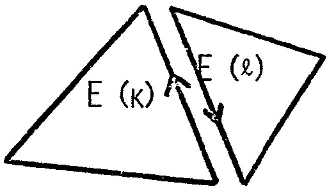
# CONDITIONS

C 1) THE TRIANGLE VERTICES ARE SPECIFIED IN COUNTER CLOCKWISE ORDER



C 2) EITHER THE ITH EDGE OF E (K) IS THE ONLY EDGE JOINING ITS END POINTS (BOUNDARY ELEMENT)

OR THERE IS EXACTLY ONE ELEMENT, E (ℓ) HAVING THE SAME EDGE. IN THIS LATTER CASE, THE DIRECTIONS OF THIS LINE SEGMENT AS EDGES OF E (K) AND E (ℓ) MUST BE DIFFERENT.



C 3) NO BOUNDARY EDGE INTERSECTS MORE THAN ONE ELEMENT, EXCEPT AT ITS END POINTS.

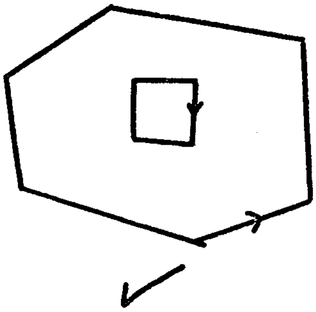
C 4) A VERTEX CAN HAVE AT MOST ONE BOUNDARY EDGE DIRECTED AWAY FROM IT.

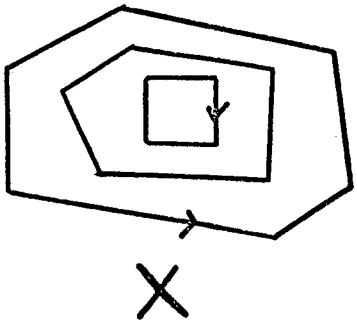
# IMPLICATIONS

1) Mesh boundary edges form a set of disjoint, oriented, simple closed curves

$$C_1, \quad C_2, \quad \ldots, \quad C_K \quad \equiv \quad \text{MESH BOUNDARY CURVES}$$

2) Each curve of bounded interior defines a connected region. The boundary of this region is composed of mesh boundary curves





(Assume 1 curve of bounded interior - $C_1$)

$$\text{DEFINE} \quad R \equiv \bigcap_{I=1}^{K} (\text{INTERIOR OF } C_I)$$

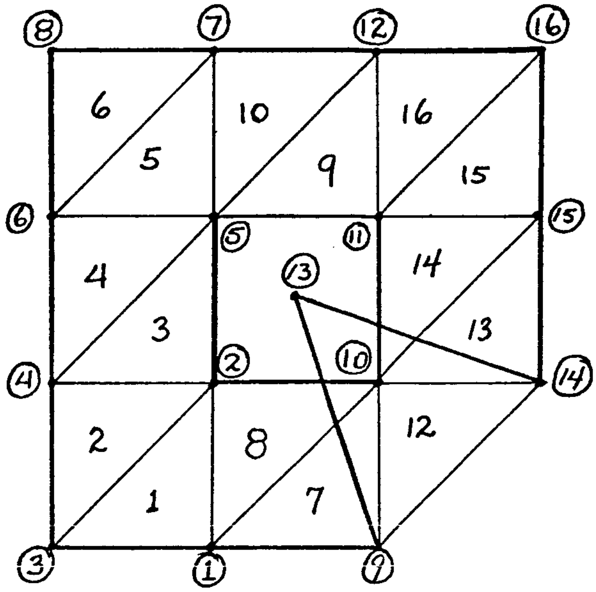
(connectivity $K$)

3) $$R = \bigcup_{J=1}^{N_E} E(J)$$

4) If $P \in R$, $P$ is not an element edge

$\Longrightarrow$ $P$ lies in exactly one element.

Small Example Invalid Mesh on Hollow Square



Coordinates of vertex 13 changed to (2.5, 2.5)

Section of Mesh Verification Algorithm Detailed Error Report

```
 MESH VERIFICATION ERROR

   INTERSECTING BOUNDARY EDGES -
   EDGE FROM VERTEX 13 AT ( 2.50, 2.50)   TO VERTEX  9 AT ( 3.00, 1.00)
   EDGE FROM VERTEX  2 AT ( 2.00, 2.00)   TO VERTEX 10 AT ( 3.00, 2.00)

 MESH VERIFICATION ERROR

   INTERSECTING BOUNDARY EDGES -
   EDGE FROM VERTEX 14 AT ( 4.00, 2.00)   TO VERTEX 13 AT ( 2.50, 2.50)
   EDGE FROM VERTEX 10 AT ( 3.00, 2.00)   TO VERTEX 11 AT ( 3.00, 3.00)

  FROM BDSCAN, NO. OF BOUNDARY CURVES=    2

 MESH VERIFICATION ERROR

   ELEMENT  11 APPEARS TO HAVE VERTICES LISTED IN WRONG ORDER
           X=    3.000000E 00    Y=    1.000000E 00
           X=    2.500000E 00    Y=    2.500000E 00
           X=    4.000000E 00    Y=    2.000000E 00
           DET =   -2.000000E 00

MESH CHECK ENCOUNTERED      3 ERRORS
```

# GRID GENERATION
# FOR
# TWO-DIMENSIONAL FINITE
# ELEMENT FLOWFIELD
# COMPUTATION

KENNETH E. TATUM
ENGINEER, TECHNOLOGY, AERODYNAMICS

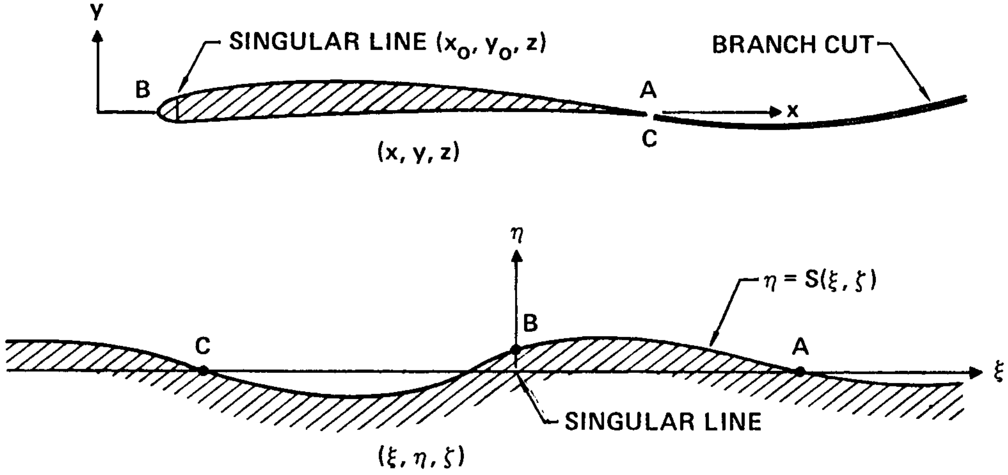
MCDONNELL AIRCRAFT COMPANY
MCDONNELL DOUGLAS CORPORATION
ST. LOUIS, MISSOURI

*To facilitate development of the finite element method for fluid dynamics problems a 2-D mesh generation scheme has been developed with the emphasis on versatility and independence of the finite element solution algorithm to be employed. No effort has been expended to maintain grid line orthogonality since the finite element method has no such requirement. The method consists of sequences of shearings and conformal maps with upper and lower surfaces handled independently to allow sharp leading edges. The method will generate meshes of triangular or quadrilateral elements. Thus, with certain additional constraints of smoothness and near-orthogonality, a quadrilateral mesh could be generated for a finite volume type method. Finally, solutions obtained by the MCAIR finite element full potential flow program on sample meshes are shown to illustrate their usefulness.*

# FIGURE 2
# TYPICAL ANALYTIC TRANSFORMATION
## PARABOLIC PLUS SHEARING

$$\xi + i\eta = \left\{[x - x_o(z) + iy - iy_o(z)]/t(z)\right\}^{1/2}$$

$$\zeta = z$$

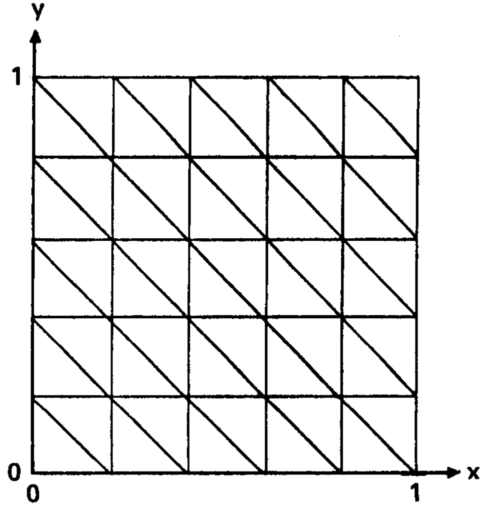


AND THE SHEARING TRANSFORMATION

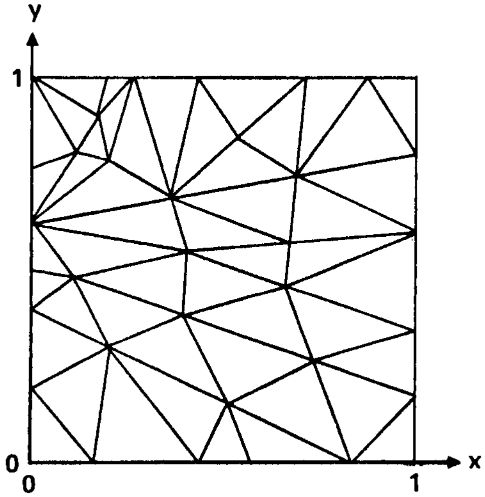
$$X = \xi \qquad Y = \eta - S(\xi, \zeta) \qquad Z = \zeta$$

*The parabolic tranformation shown is a typical method used to generate body fitted coordinate meshes for 2-D flowfield computations. Precise transformation Jacobians must be defined relating the uniform cartesian computational grid to the physical body-conforming coordinate grid. Computations are performed by Finite Difference Methods (FDM) in the cartesian coordinate space with determinants of the Jacobians appearing as added coefficients in the difference equations. Simple analytic transformations, even if multiple, cause little increase in complexity of the equations. However for complex body shapes numeric transformation techniques must be employed requiring Jacobian matrices to be computed for each grid cell. These matrices, often approximated, must be stored within the computer or recomputed for successive iterations of nonlinear systems. Either technique is costly.*

# FIGURE 3
# POSSIBLE FINITE ELEMENT MESH TRIANGULATION ON UNIT SQUARE



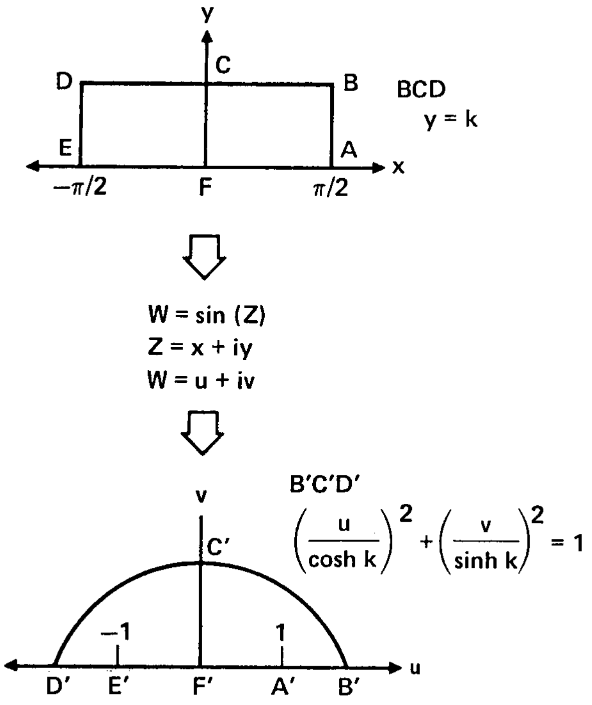
36 NODES
50 ELEMENTS



36 NODES
48 ELEMENTS

McDonnell Aircraft Company (MCAIR) is studying the Finite Element Method (FEM) as a method which might eliminate, or drastically reduce, the cost associated with transformation Jacobians. The FEM is equally suited to uniform cartesian meshes or irregular, highly non-orthogonal meshes. Two distinctly different FEM meshes (triangulations) of the unit square are shown in Figure 3. Each mesh contains 36 nodes and are equally usable even though a specific problem may indicate the desirability of one over the other. Computations may thus be performed directly in physical space on body-fitted grids generated independently of orthogonality constraints. Only the physical nodal coordinates and the relationships of nodes to elements must be stored.

# FIGURE 4
## COMPLEX SINE CONFORMAL MAPPING

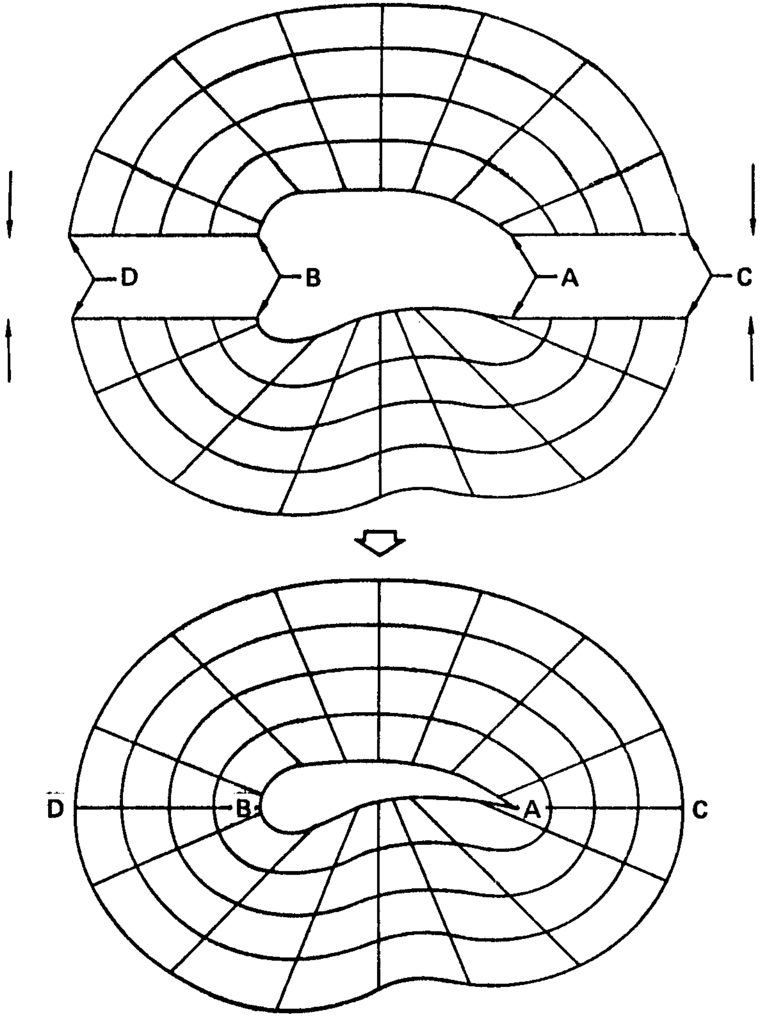


$$W = \sin(Z)$$
$$Z = x + iy$$
$$W = u + iv$$

$$\left(\frac{u}{\cosh k}\right)^2 + \left(\frac{v}{\sinh k}\right)^2 = 1$$

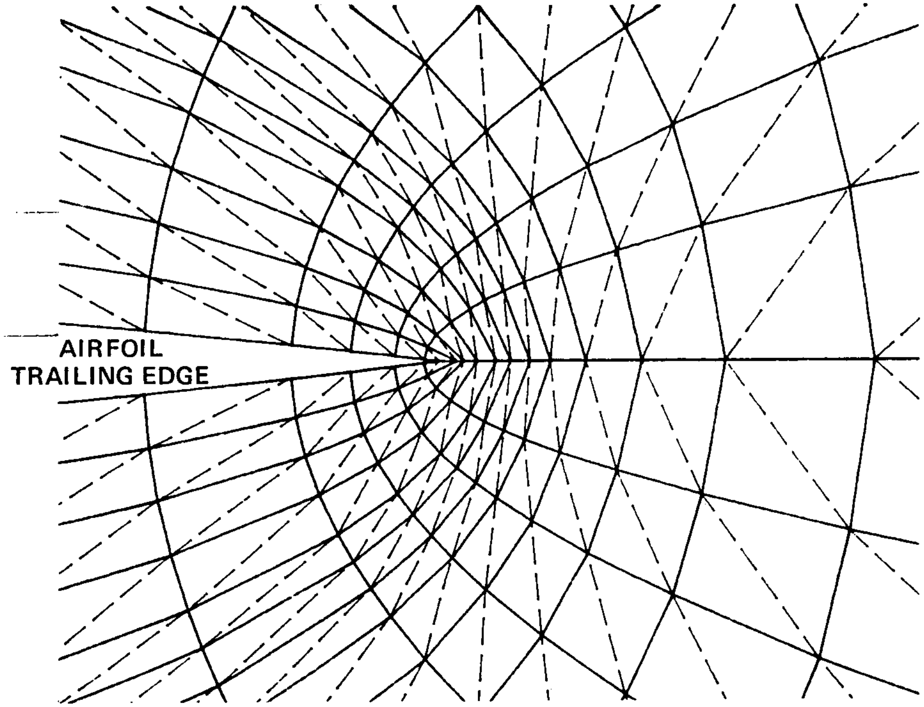
*Grid generation for a FEM computation may be performed by many means. While conformal mappings of simple, highly regular grids are not necessary from the standpoint of maintaining orthogonality, they are useful in producing grids with simple relationships between nodes and elements. Accordingly the current MCAIR technique is based on a conformal (sine) mapping of a rectangular region to a semi-oval region. A sequence of shearing and stretching transformations, both prior to and subsequent to the conformal mapping, shape line E′F′A′ to that of one surface of the airfoil, either upper or lower. The number of, or localization of, the shearings is entirely unimportant as long as they may be programmed easily.*

# FIGURE 5
# MATCHING OF UPPER AND LOWER MESH REGIONS



*The general method described by Figure 4 is used twice to form two mesh regions as shown in Figure 5, one about the upper surface and one about the lower surface. The airfoil is situated with the forward-most and aft-most points at $y = 0, x = \pm 1$ and the two regions are designed to match along the line $y = 0, | x | \geqslant 1$. Points along the line AC are doubly specified thus creating a cut across which wake/circulation boundary conditions may be applied. Points along the matching line BD are merged and no boundary is considered there in the final mesh.*

FIGURE 6
TRIANGULATION OF FIELD BY DIVIDING QUADRILATERALS ALONG APPROPRIATE DIAGONAL
DASHED LINES = DIAGONAL



*The actual computation of nodal coordinates has been automated in a Fortran computer code for an arbitrary airfoil either specified analytically or by discrete points. Program inputs allow the exact specification of the desired mesh spacing along the body surface as well as the relative spacings normal to the surface. The final field is triangulated as shown in Figure 6 by dividing quadrilaterals along a diagonal, with the diagonal direction varying between regions of the mesh in such a way as to prevent the conformal map from collapsing a triangle to zero area. Triangular elements are desired only because of the simplicity of finite element integration over such regions. However, if desired, quadrilateral elements are obviously generated quite as readily by the scheme.*

# FIGURE 7
## 16.3% NLR 7301 AIRFOIL
### 72 × 17 MESH

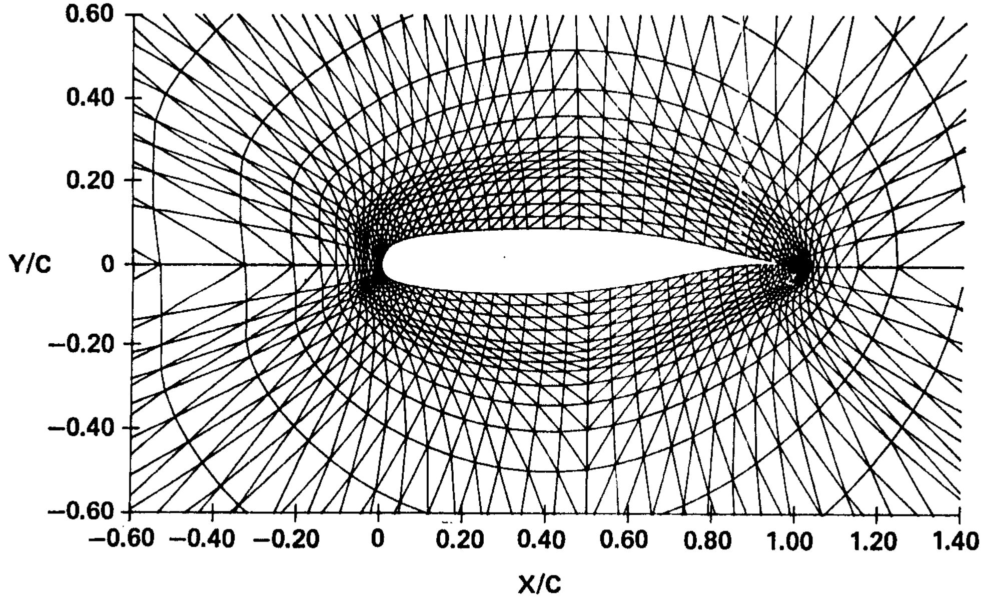


*Figure 7 shows a final resultant mesh generated about a modern supercritical airfoil, the 16.3% thick NLR 7301 airfoil, with the coordinate system scaled by the airfoil chord. The mesh consists of 72 elements bounding the airfoil surface with 17 rows of elements extending outward from the surface. A total of 1314 nodes define the 2448 elements. Neither the bluntness of the leading edge region nor the reverse curvature of the aft lower surface create any difficulties for the method.*

## FIGURE 8
## 6% SYMMETRIC BICONVEX AIRFOIL
### 72 × 17 MESH

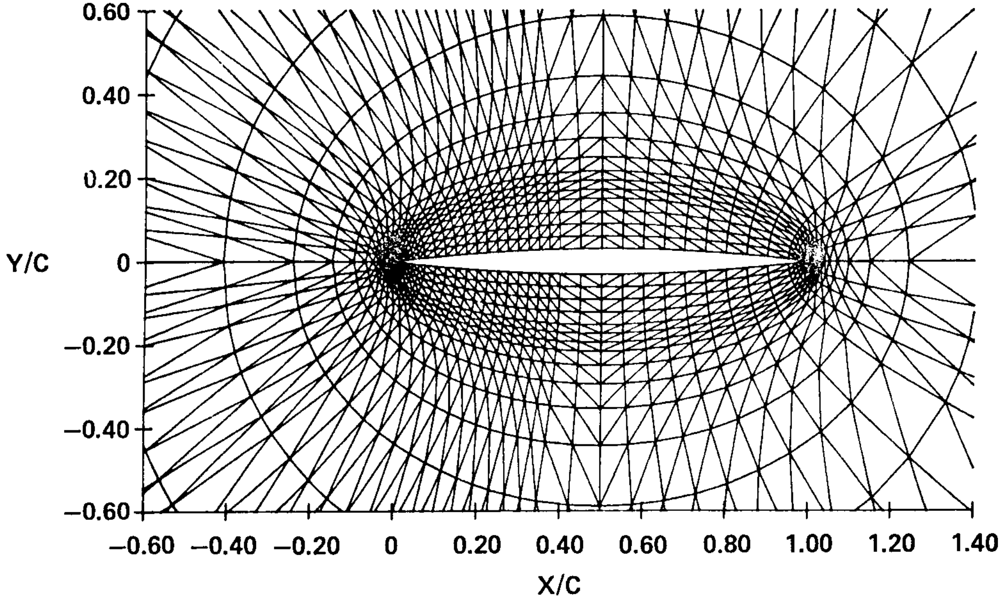


*Figure 8 shows a sample mesh about an airfoil with opposite extremes to that of Figure 7. The airfoil is a thin (6%) symmetric Biconvex section with sharp leading and trailing edges. The sharpness of the leading edge presents no difficulties for the method due to the independent handling of upper and lower surfaces. No singular point ($x_o$, $y_o$), as needed for example by the parabolic transformation illustrated in Figure 2, exists for this type of leading edge. Thus, any solution procedure depending on a singularity point unwrapping transformation will fail on this airfoil.*

# FIGURE 9
# 6% SYMMETRIC BICONVEX AIRFOIL
## LEADING EDGE

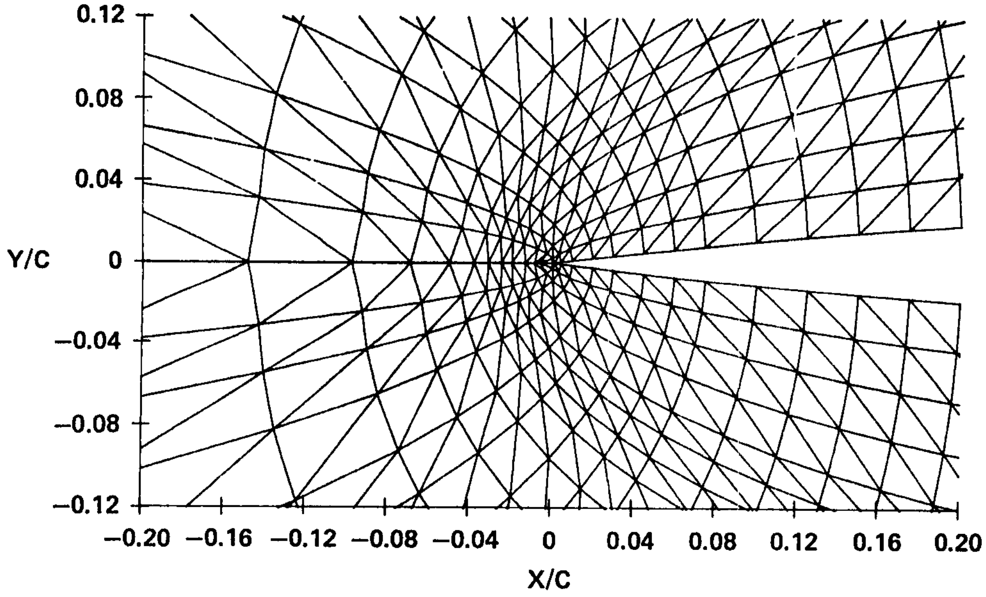
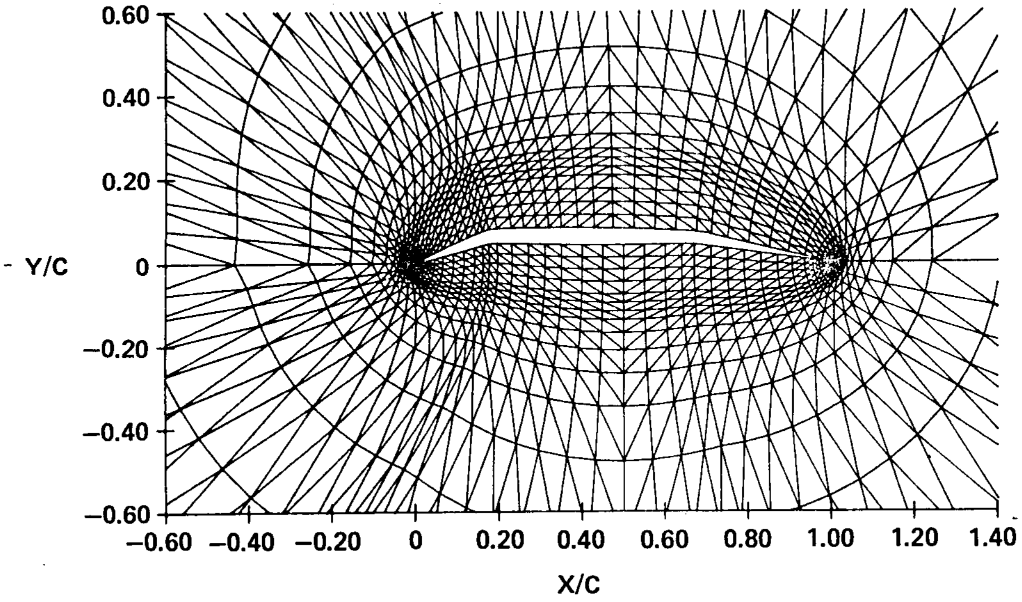


*Figure 9 is an enlargement of the leading edge region of the mesh in Figure 8. Some local stretchings of the mesh have been automatically performed by the computer code to prevent some elements from becoming too small or thin. The stretchings may distort the smooth variation of elements around the leading edge but in reality increase the potential for obtaining an accurate finite element solution on the mesh due to the maintaining of smaller aspect ratios (maximum/minimum dimension of triangle) of individual elements. Another constraint on the elements necessitating some local stretching is that the magnitude of the area of the smallest element not be too small relative to significant digit resolution of the computer on which calculations are to be performed.*

# FIGURE 10
## SUPERSONIC FIGHTER AIRFOIL (20° AND 10° FLAPS)
### 72 × 17 MESH



*The use of shearing transformations not restricted to maintaining orthogonality of the grid allows the creation of grids about sharp corners. Figure 10 shows a grid about a supersonic fighter airfoil section with both leading and trailing edge flaps deflected. The discontinuous surface slopes at the hinge lines might create numerical singularities in methods which attempt to maintain orthogonal grids. The MCAIR technique however has no such difficulties. The computer program also allows element spacings to be user specified to facilitate bunching of elements around the flap hinges where high velocity gradients are to be expected in the FEM solution.*

## FIGURE 11
## NLR SYMMETRIC SHOCK-FREE AIRFOIL
### 36 × 17 MESH

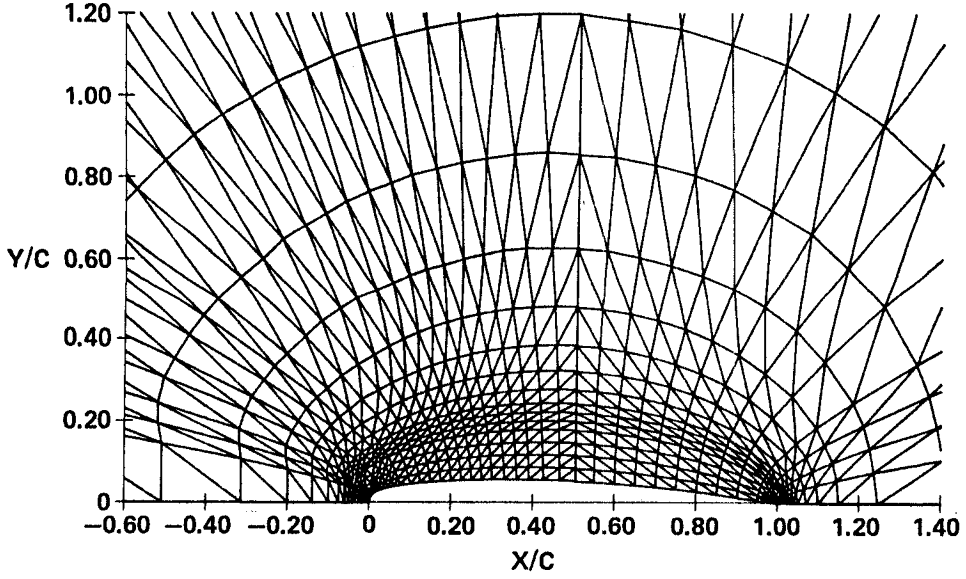
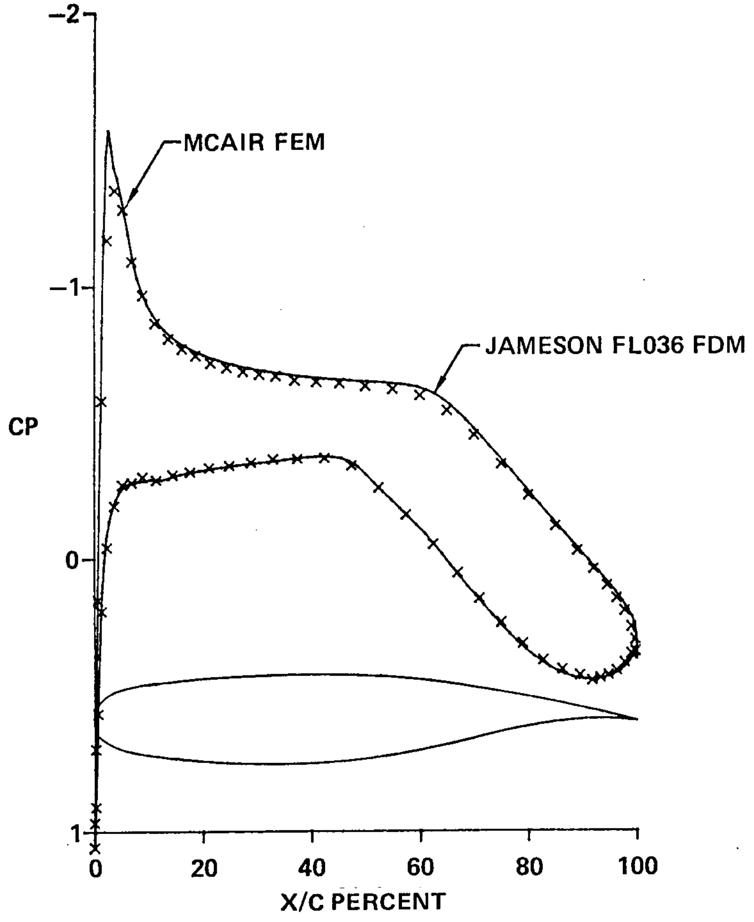


*Figure 11 is an example of a half-plane mesh about a symmetric NLR shock-free airfoil design. The independent handling of upper and lower airfoil surfaces allows this type of mesh to be generated very simply. Flow solutions about symmetric airfoils at zero angle-of-attack may thus be obtained at half the expense or with double the nodal density but no additional cost. Such solutions are important in fundamental research and also for comparison of full potential flow solutions with small perturbation solutions. These small perturbation solutions are most strictly valid at small, or zero, angle-of-attack on thin or sharp nosed airfoils such as the biconvex section shown in Figure 8.*

## FIGURE 12
## 16.3% NLR 7301 AIRFOIL
### ALPHA = 0.391° MACH 0.502



*Figures 12-14 illustrate the use of the meshes shown in previous figures by the current MCAIR FEM full potential flow program. Figure 12 compares a FEM solution at a moderate subsonic Mach number (0.5) and small angle-of-attack to the solution by a modern state-of-the-art Finite Difference Method (FDM) program. The comparison is good even though the FDM grid was much denser than the FEM mesh.*

## FIGURE 13
## SUPERSONIC FIGHTER AIRFOIL (20° AND 10° FLAPS)
### ALPHA = 1.0° MACH 0.5

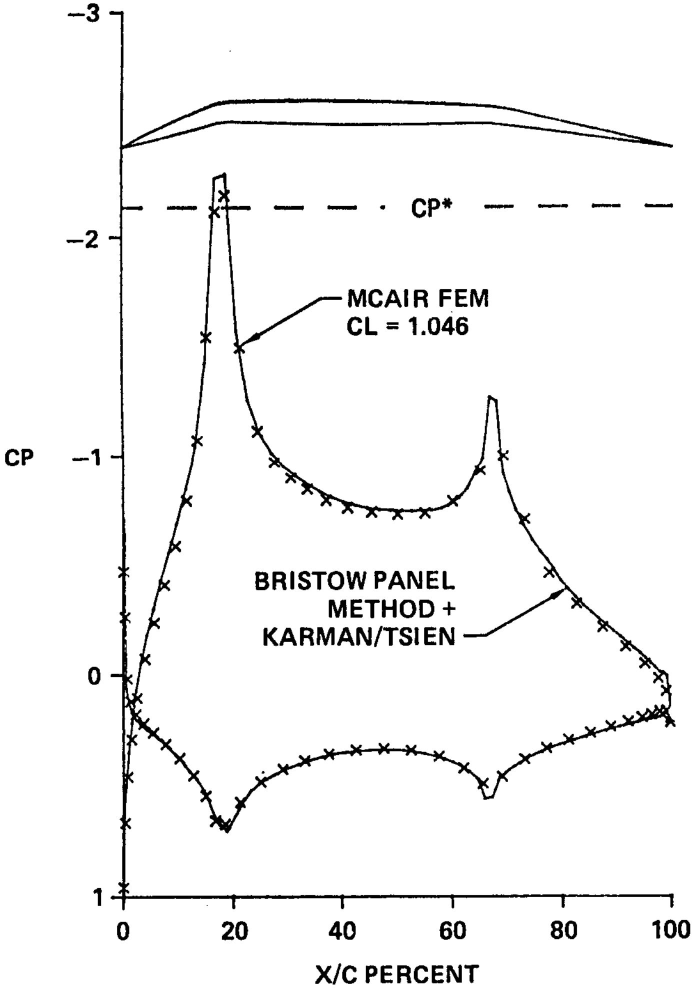


*Figure 13 compares solutions for the supersonic fighter airfoil with 20° leading edge and 10° trailing edge flaps. Since no finite difference program was available which would compute flows about sharp leading edges and abrupt hinge lines, a modern technology Panel Method program, the Bristow Multielement Airfoil Analysis and Design (MAAD) program, was employed for comparison. A Karman-Tsien compressibility correction was applied to obtain a Mach 0.5 solution. Agreement is good even in regions of pressure spikes, however the inexpensive panel method employed approximately double the panel (solution node) density of the FEM and was able to more accurately resolve the pressure peaks.*

# FIGURE 14
## NLR SYMMETRIC AIRFOIL
### ALPHA = 0° MACH 0.786

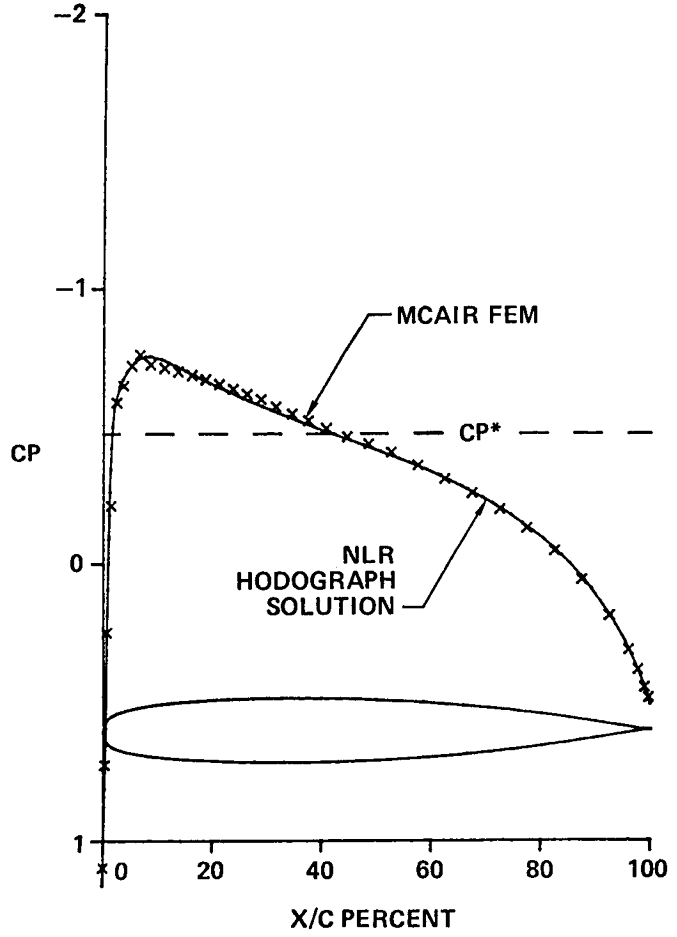
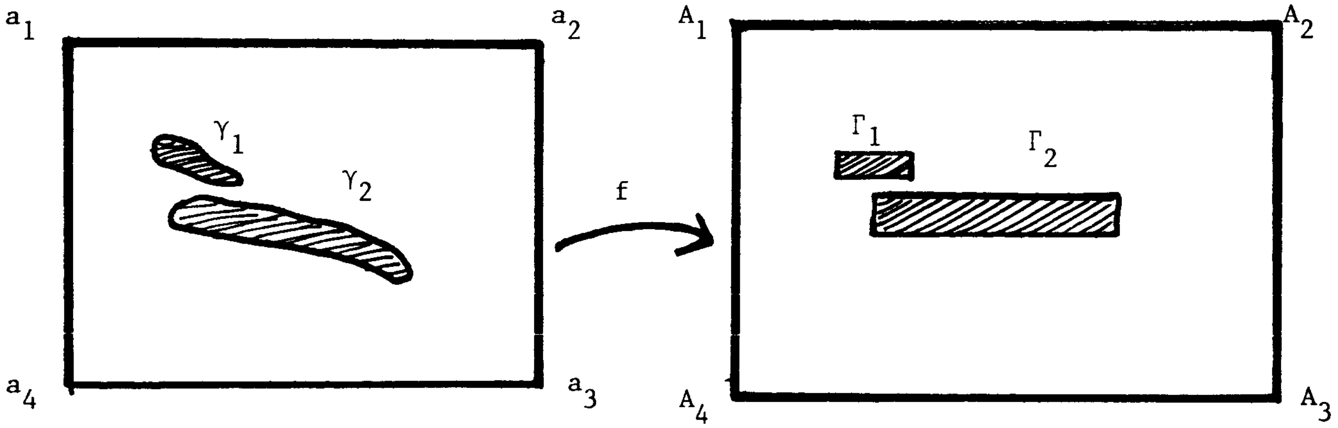


***Figure 14 illustrates the use of a half-plane mesh, specifically that shown in Figure 10. The solution was obtained in the fundamental research on adapting the FEM to non-subsonic flowfields where the governing differential equations are of mixed elliptic/hyperbolic type. Using the artificial density concept of Holst the MCAIR FEM was able to produce this solution on a shock-free airfoil which agrees reasonably well with the theoretical hodograph solution. The versatility of the mesh generator, both in technique and program, greatly facilitates the FEM research into fundamental computational methods and in applied fluid mechanics.***

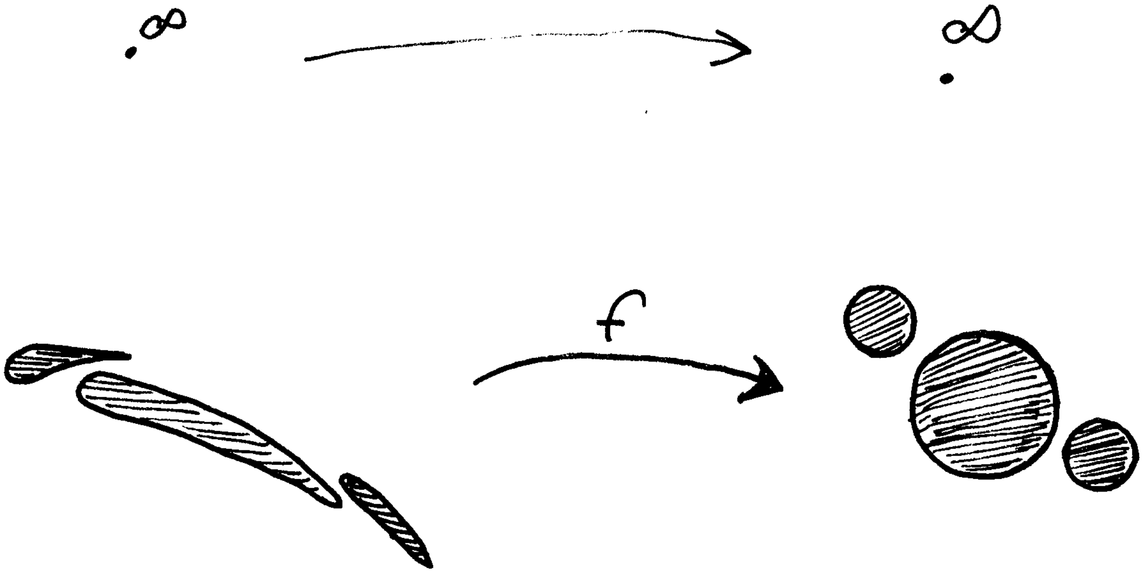
# Conformal Mappings of Multiply Connected Regions onto Regions with Specified Boundary Shapes

by Andrew N. Harrington
School of Mathematics
Georgia Institute of Technology



f conformal $\quad f(a_j) = A_j,\ \gamma_j \xrightarrow{f} \Gamma_j$

The author has developed and implemented a numerical procedure to compute the conformal mapping of a given n-tuply connected region onto a region with any specified boundary shapes and with several possible normalizations. If we start with a region whose outer boundary is a rectangle, we may arrange that the outer boundary of the image region is also a rectangle, and the vertices map to vertices. We may chose the inner boundaries to map to rectangles or to any other shapes.



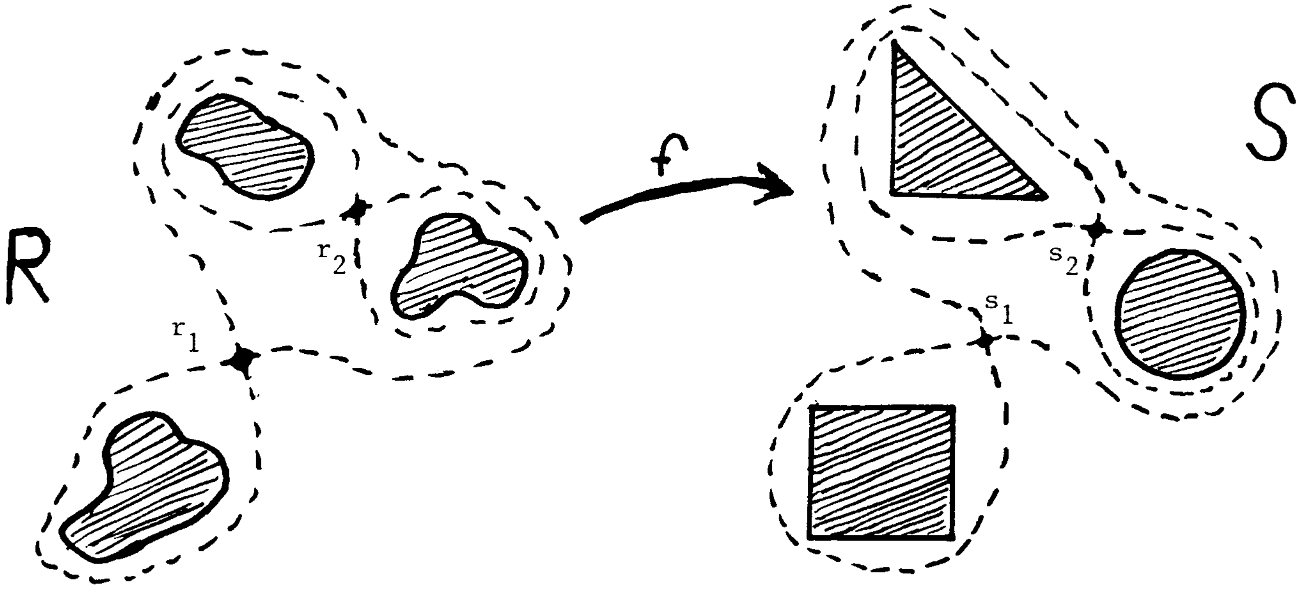
We may also consider unbounded regions and find a mapping normalized at $\infty$ $z + O(1/z)$. We may chose the boundaries of the image region to be circles or any other shapes.

Method

Though we may specify boundary shapes and orientations arbitrarily, the proper translation and magnification parameters must be calculated to determine the image domain and the mapping. For example, in order to find a conformal mapping between n-tuply connected regions R and S containing $\infty$ with $f(\infty) = \infty$, we must satisfy conditions on $G_R$ and $G_S$, the analytic completions of the Green's functions for R and S with pole at $\infty$. We must have

$$G_R(r_j) = G_S(s_j) \qquad j = 1, 2, \ldots n-1$$

where $r_j$ and $s_j$, $j = 1, 2, \ldots n-1$, are the critical points for $G_S$ and $G_R$ labeled in the figure. Using Symm's method to approximate Green's functions one may easily calculate the appropriate parameters. Then $G_S(f(z)) = G_R(z)$.



The dotted curves are the level curves of Re $G_R$ and Re $G_S$ which branch at the critical points.

# FINITE DIFFERENCE GRID GENERATION BY MULTIVARIATE BLENDING FUNCTION INTERPOLATION*

by

Peter G. Anderson
Lawrence W. Spradley

Computational Fluid Dynamics Section
Lockheed-Huntsville Research & Engineering Center
Huntsville, AL 35807

*This work was supported, in part, by NASA-Langley Contracts NAS1-15341, 15783 and 15795

# ABSTRACT

The General Interpolants Method (GIM) code solves the multidimensional Navier-Stokes equations for arbitrary geometric domains. The geometry module in the GIM code generates two- and three-dimensional grids over specified flow regimes, establishes boundary condition information and computes finite difference analogs for use in the GIM code numerical solution module. The technique can be classified as an algebraic equation approach.

The geometry package uses multivariate blending function interpolation of vector-values functions which define the shapes of the edges and surfaces bounding the flow domain. By employing blending functions which conform to the cardinality conditions the flow domain may be mapped onto a unit square (2-D) or unit cube (3-D), thus producing an intrinsic coordinate system for the region of interest. The intrinsic coordinate system facilitates grid spacing control to allow for optimum distribution of nodes in the flow domain.

The GIM formulation is not a finite element method in the classical sense. Rather, finite difference methods are used exclusively but with the difference equations written in general curvilinear coordinates. Transformations are used to locally transform the physical planes into regions of unit cubes. The mesh is generated on this unit cube and local metric-like coefficients generated. Each region of the flow domain is likewise transformed and then blended via the finite element formulation to form the full flow domain. In order to treat "completely-arbitrary" geometric domains, different transformation functions can be employed in different regions. We then transform the blended domain to physical space and solve the Cartesian set of equations for the full region. The geometry part of the problem is thus treated much like a finite element technique while integration of the equations is done with finite difference analogs.

# BUILDING BLOCK CONCEPT

The development is done in local curvilinear intrinsic coordinates based on the following concepts:

- Analytical regions such as rectangles, spheres, cylinders, hexahedrals, etc., have intrinsic or natural coordinates.
- Complex regions can be subdivided into a number of smaller regions which can be described by analytic functions. The degenerate case is to subdivide small enough to use very small straight-line segments.
- Intrinsic curvilinear coordinate systems result in constant coordinate lines throughout a simply connected, bounded domain in Euclidean space.
- The intersection of the lines of constant coordinates produce nodal points evenly spaced in the domain.
- Intrinsic curvilinear coordinate systems can be produced by a univalent mapping of a unit cube onto the simply connected bounded domain.

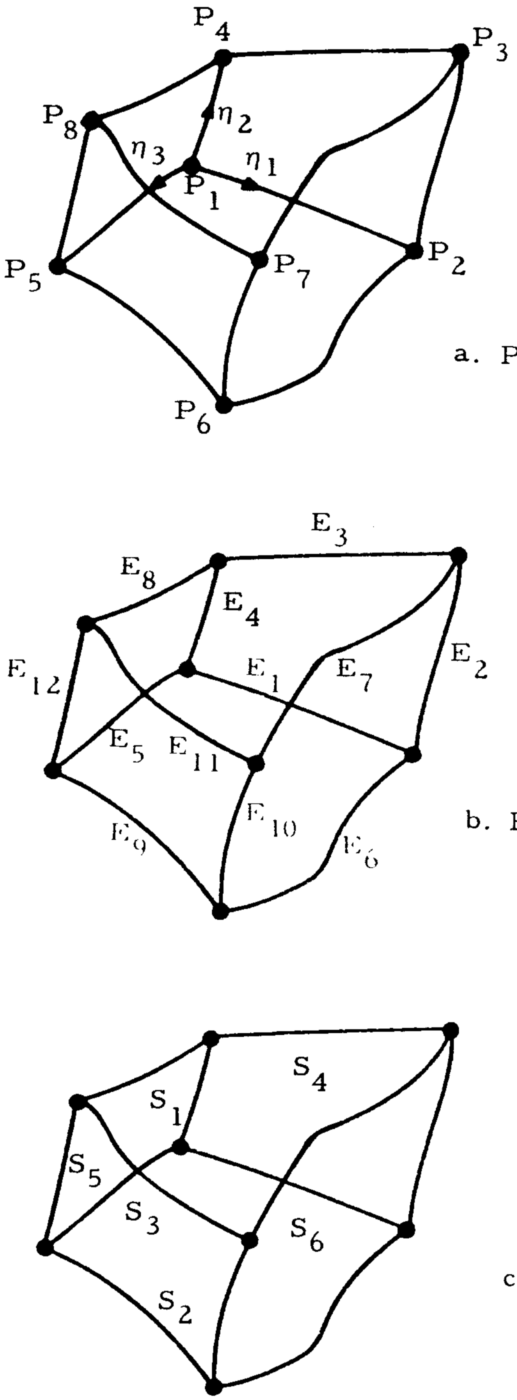
Thus, if a transformation can be found which will map a unit cube univalently onto a general analytical domain, then any complex region can be piecewise transformed and blended using general interpolants.

Consider the general hexahedral configuration shown. The local intrinsic coordinates are $\eta_1$, $\eta_2$, $\eta_3$ with origin at point $P_1$. The shape of the geometry is defined by

- Eight corner points, $\overline{P}_i$
- Twelve edge functions, $\overline{E}_i$
- Six surface functions, $\overline{S}_i$

This shape is then fully described if the edges and surfaces can be specified as continuous analytic vector functions $\overline{S}_i(x, y, z)$, $\overline{E}_i(x, y, z)$.

# BUILDING BLOCK CONCEPT



a. Point Designations

b. Edge Designations

c. Surface Designations

# GENERAL INTERPOLANT FUNCTION

Based on the work of Gordon and Hall we have developed a general relationship between physical Cartesian space and local curvilinear intrinsic coordinates. This relation is given by the general trilinear interpolant shown on the adjacent figure.

In this equation, $\underline{\overline{X}}$ vector is the Cartesian coordinates

$$\underline{\overline{X}}(\eta_1, \eta_2, \eta_3) = \begin{bmatrix} x(\eta_1, \eta_2, \eta_3) \\ y(\eta_1, \eta_2, \eta_3) \\ z(\eta_1, \eta_2, \eta_3) \end{bmatrix}$$

and $S_i$, $E_i$ are the vector functions defining the surfaces and edges, respectively, and $\overline{P}_i$ are the (x, y, z) coordinates of the corner points. Edge and surface functions that are currently included in the GIM code are the following:

- EDGES (Combinations of up to Five Types)

  Linear Segment
  Circular Arc
  Conic (Elliptical, Parabolic, Hyperbolic)
  Helical Arc
  Sinusoidal Segment

- SURFACES (Bounded by Above Edges)

  Flat Plate
  Cylindrical Surface
  Edge of Revolution

This library of available functions is simply called upon piecewise via input to the computer code.

With this transformation, any point in local coordinates $\eta_1$, $\eta_2$, $\eta_3$ can be related to global Cartesian coordinates x, y, z. Likewise any derivatives of functions in local coordinates can be related to that derivative in physical space.

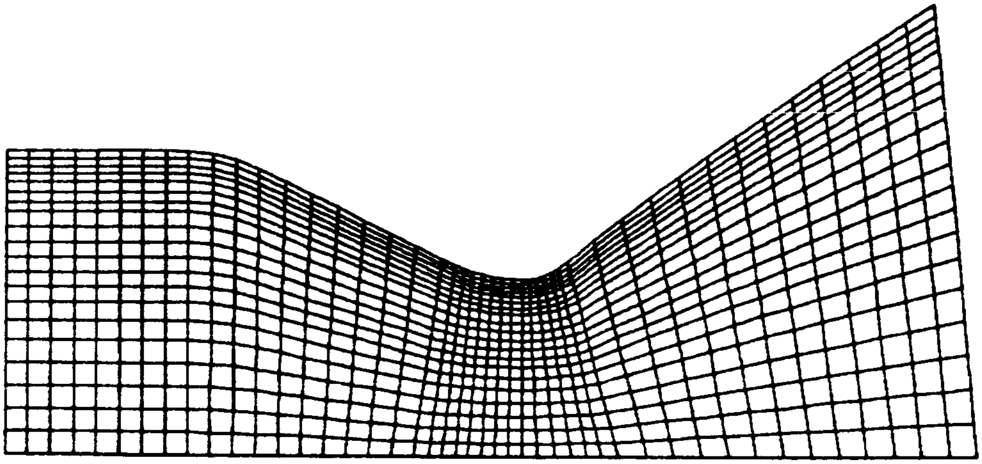
# GENERAL INTERPOLANT FUNCTION

$$
\begin{aligned}
&\underline{\overline{X}}\,(\eta_1,\eta_2,\eta_3) \;= \\
&(1-\eta_1)\,\vec{S}_5 + \eta_1\,\vec{S}_6 + (1-\eta_2)\,\vec{S}_2 + \eta_2\,\vec{S}_4 \\
&+ (1-\eta_3)\,\vec{S}_1 + \eta_3\,\vec{S}_3 \\
&- (1-\eta_1)\,(1-\eta_2)\,\vec{E}_5 - (1-\eta_1)\,\eta_2\,\vec{E}_8 - \eta_1\,(1-\eta_2)\,\vec{E}_6 \\
&- \eta_1\,\eta_2\,\vec{E}_7 - (1-\eta_1)\,(1-\eta_3)\,\vec{E}_4 - (1-\eta_1)\,\eta_3\,\vec{E}_{12} \\
&- \eta_1\,(1-\eta_3)\,\vec{E}_2 - \eta_1\,\eta_3\,\vec{E}_{10} - (1-\eta_2)\,(1-\eta_3)\,\vec{E}_1 \\
&- (1-\eta_2)\,\eta_3\,\vec{E}_9 - \eta_2\,(1-\eta_3)\,\vec{E}_3 - \eta_2\,\eta_3\,\vec{E}_{11} \\
&+ (1-\eta_1)\,(1-\eta_2)\,(1-\eta_3)\,\vec{P}_1 + (1-\eta_1)\,(1-\eta_2)\,\eta_3\,\vec{P}_5 \\
&+ (1-\eta_1)\,\eta_2\,(1-\eta_3)\,\vec{P}_4 + (1-\eta_1)\,\eta_2\,\eta_3\,\vec{P}_8 \\
&+ \eta_1\,(1-\eta_2)\,(1-\eta_3)\,\vec{P}_2 + \eta_1\,(1-\eta_2)\,\eta_3\,\vec{P}_6 \\
&+ \eta_1\,\eta_2\,(1-\eta_3)\,\vec{P}_3 + \eta_1\,\eta_2\,\eta_3\,\vec{P}_7
\end{aligned}
$$

# INTERNAL FLOW GRID
## (Axisymmetric Rocket Nozzle)

The grid shown was used to compute the flow in a model of the Space Shuttle engine using the GIM code. The mesh is stretched in the radial direction to cluster points near the wall and stretched axially to place points near the throat of the nozzle. Only a portion of the complete grid is shown for clarity and illustration. The grid shows the general format used by the GIM code for internal, two-dimensional flows in non-rectangular shapes.

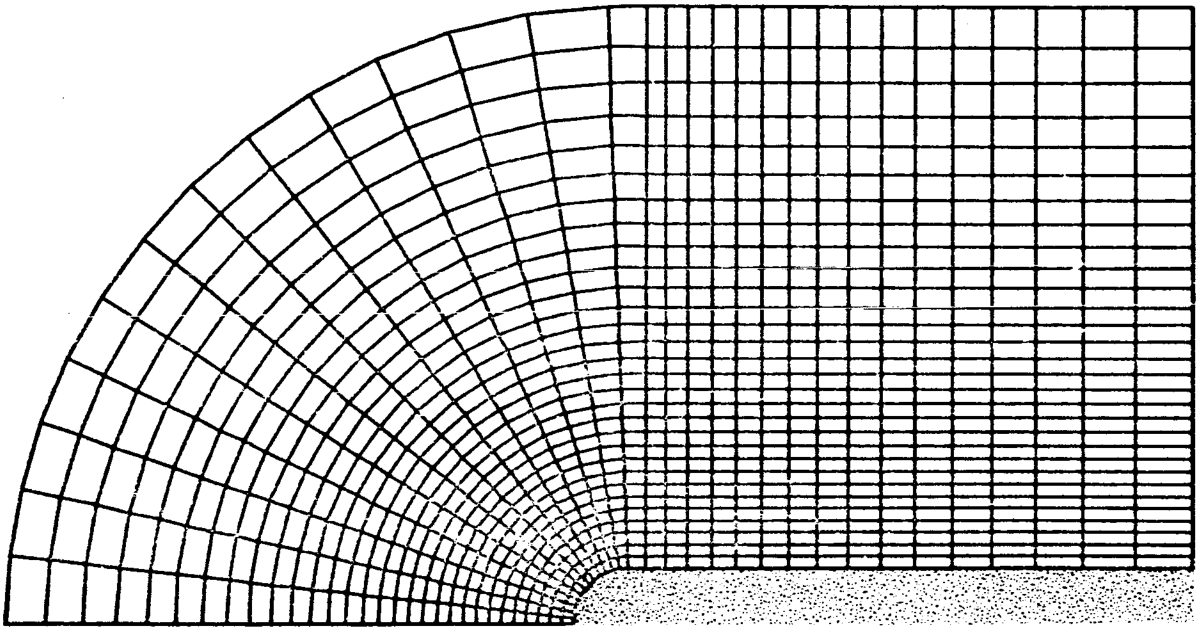
## INTERNAL FLOW GRID
## (Axisymmetric Rocket Nozzle)



# EXTERNAL FLOW GRID
# (Two-Dimensional Blunt Body Flow)

This figure shows a polar-like grid used for computing external flow over a blunt body. The body surface is treated inviscidly, and thus does not require an extremely tight mesh. The outer boundary is the freestream flow. The grid illustrates the GIM code technique for two-dimensional external flows using a polar-like coordinate system.

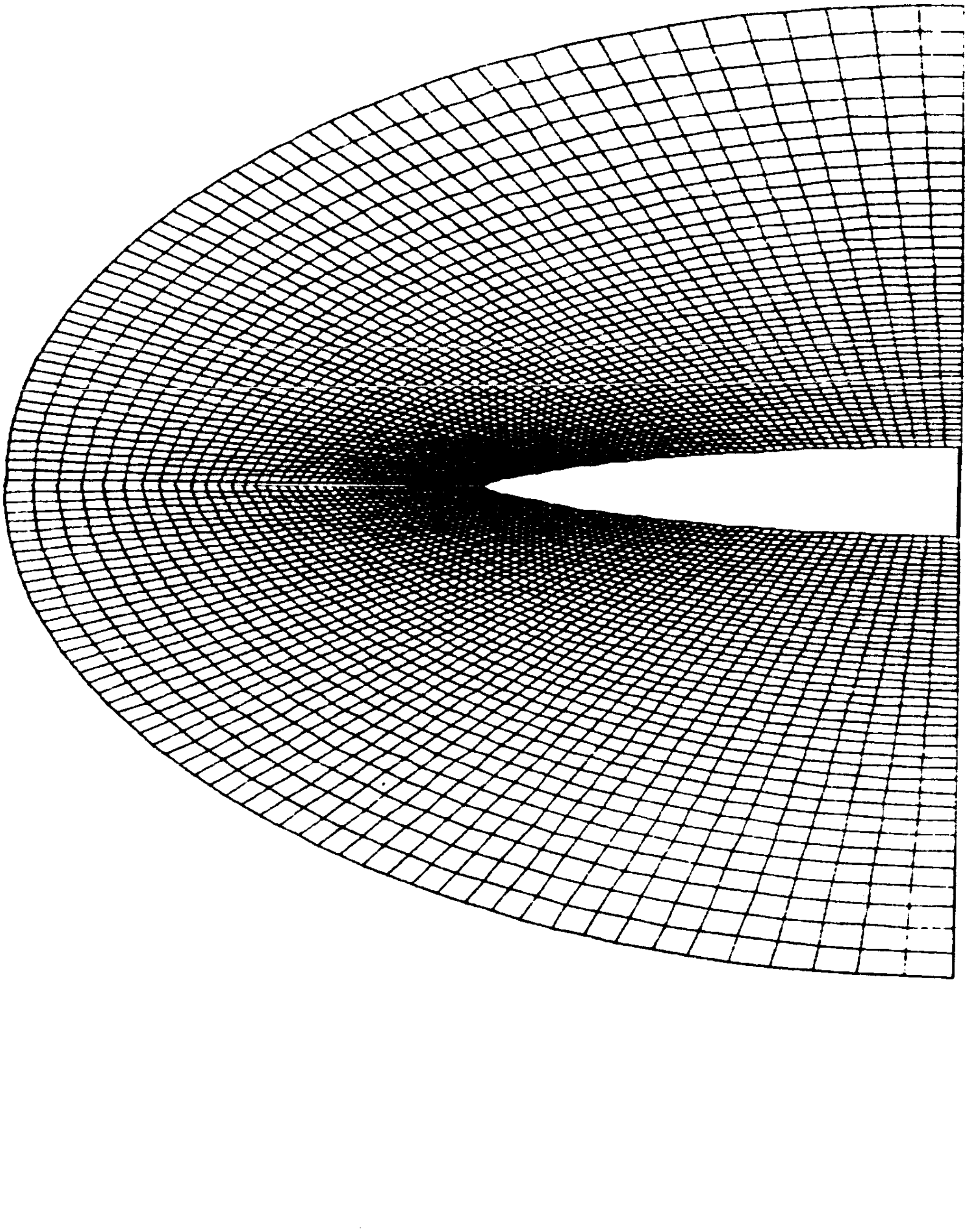
## EXTERNAL FLOW GRID
(Two-Dimensional Blunt Body Flow)



# EXTERNAL FLOW GRID
## (Non-Orthogonal Curvilinear Coordinates)

The nodal network for the external flow over an ogive cylinder illustrates the capability of the GIM code geometry package to stretch the nodal distribution. The grid is very compact in the leading edge region and greatly expanded in the far field areas. The axial points follow the body surface and could generally be called "body-oriented coordinates" in the nomenclature of the literature. The radial grid lines are not necessarily normal to the lateral lines or to the body surface. The GIM code, through its "nodal-analog" concept can operate on this general non-orthogonal curvilinear grid.

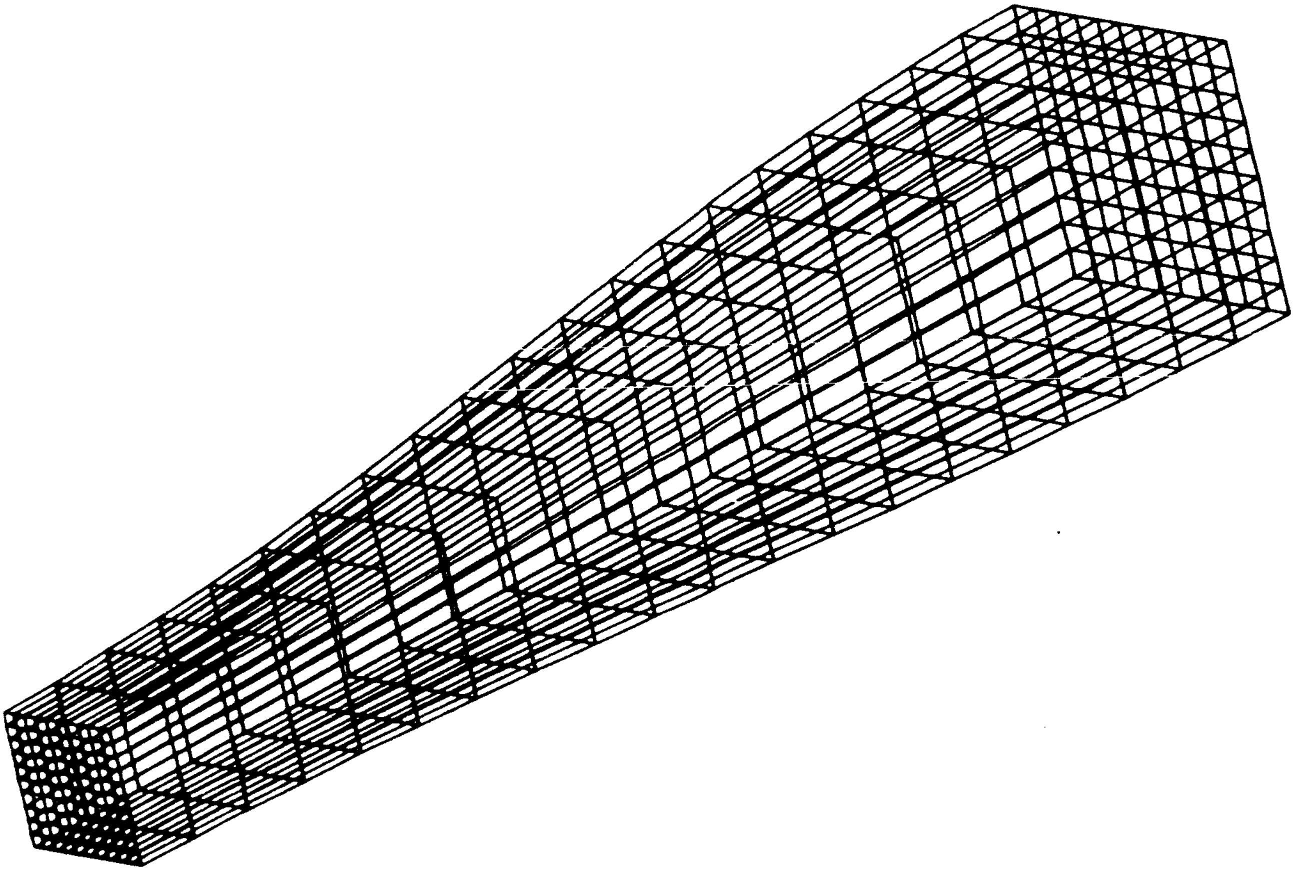
EXTERNAL FLOW GRID
(Non-Orthogonal Curvilinear Coordinates)



# THREE-DIMENSIONAL GRID
## (Simple Rectilinear Coordinates)

Supersonic flow in expanding ducts of arbitrary cross section is a common occurrence in computational fluid dynamics. This figure illustrates a simple grid for a three-dimensional duct whose cross section varies sinusoidally with the axial coordinate. The "top" wall and the "front" wall have this sinusoidal variation while the "bottom" and "back" walls are flat plates. The grid shown was used to resolve the expanding-recompressing supersonic flow including the intersection of the two shock sheets.

## THREE-DIMENSIONAL GRID
(Simple Rectilinear Coordinates)

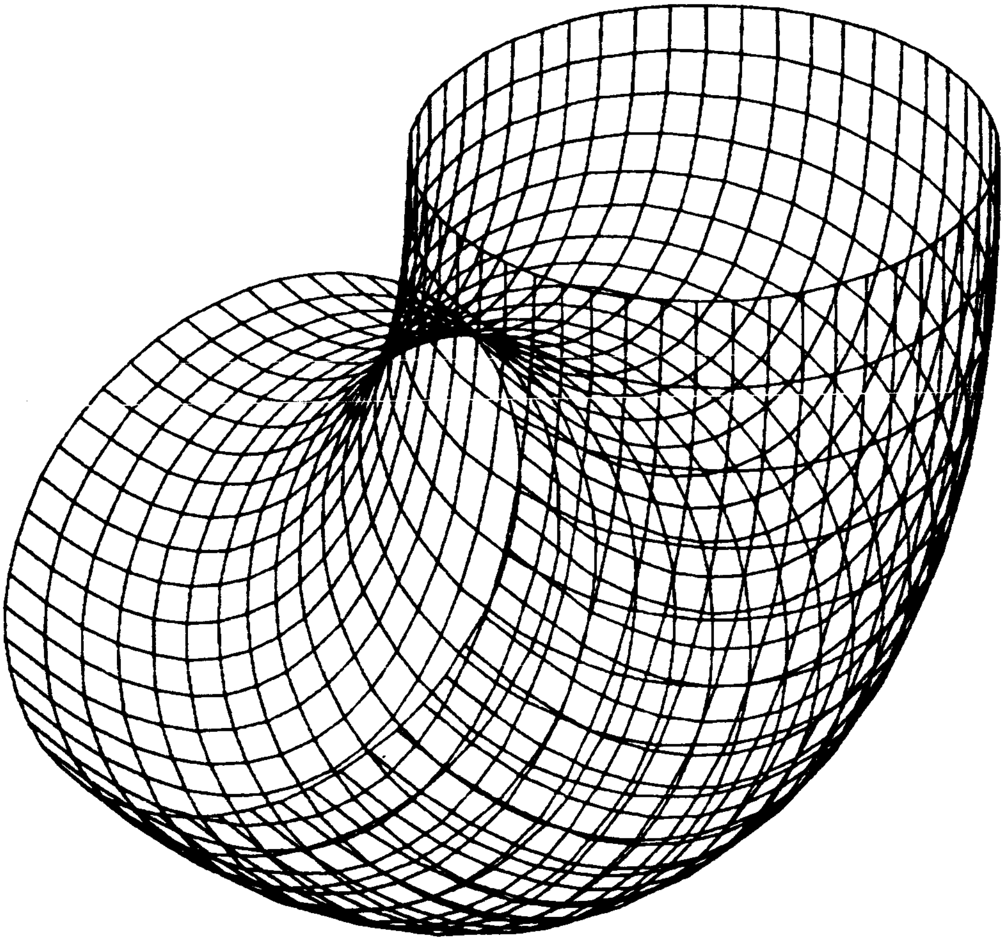


# THREE-DIMENSIONAL GRID
# (Pipe Flow in a 90 deg Elbow Turn)

There are numerous flow fields of interest which contain a sharp turn inside a smooth pipe. The GIM code has treated certain of these for application to jet deflector nozzle flow in VTOL aircraft. The portion of a grid shown in the adjacent figure was used for this calculation.

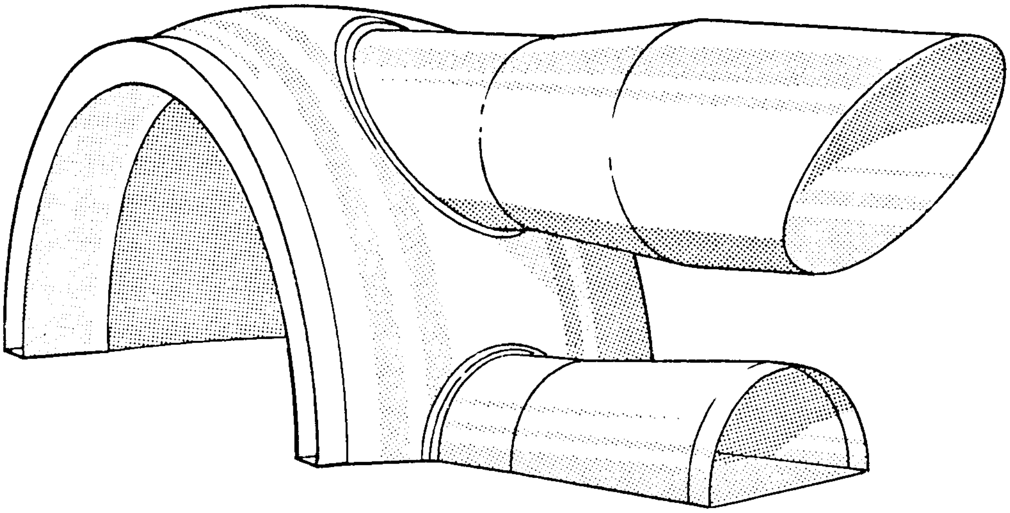
The 90 deg elbow demonstrates the capability to model three-dimensional non-Cartesian geometries. The internal nodes were emitted for clarity. The elbow grid was generated by employing edge-of-revolution surfaces with circular arc segments as the edges being revolved.

THREE-DIMENSIONAL GRID
(Pipe Flow in a 90 deg Elbow Turn)



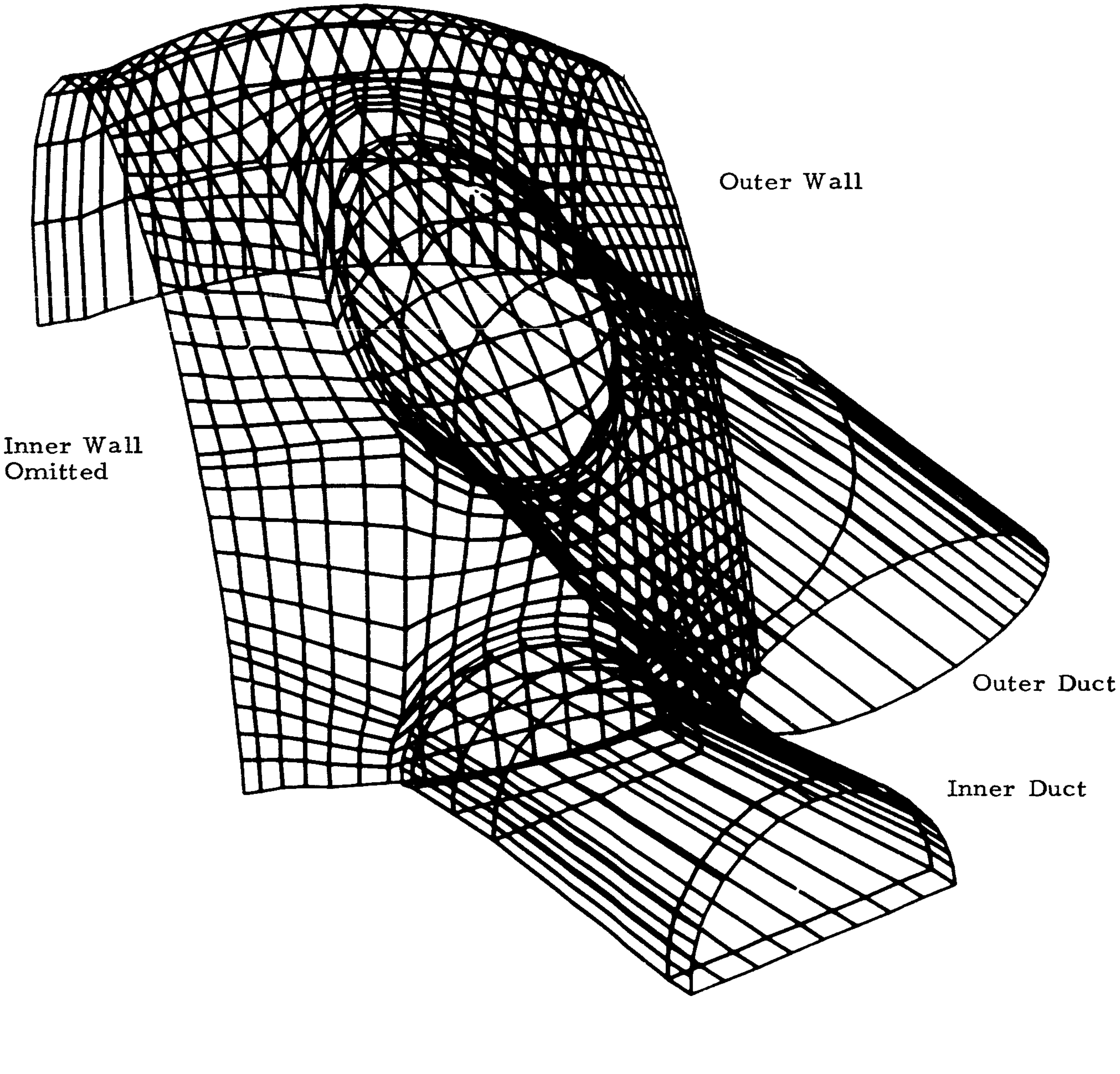
# GRID FOR SPACE SHUTTLE MAIN ENGINE
## (Hot Gas Manifold Geometry Model)

The recent problems encountered with the Space Shuttle main engine tests have resulted in a GIM code analysis of the system. The "hot gas manifold" is a portion of this analysis for the high pressure turbopump system. The grid shown in the adjacent figure was used for this calculation. Only a small number of nodes are shown for clarity; the full model consists of approximately 14,000 nodes. The extreme complexity of this geometry illustrates the necessity of using a GIM-like technique. Transforming this case to a square box computational domain is, of course, impossible. The results of the GIM code analysis agrees qualitatively with flow tests that have been run on the hot gas manifold.



Hot Gas Manifold Configuration

## GRID FOR SPACE SHUTTLE MAIN ENGINE
(Hot Gas Manifold Geometry Model)



# SUMMARY

- Finite difference grids can be generated for very general configurations by using multivariate blending function interpolation.

- The GIM code difference scheme operates on general non-orthogonal curvilinear coordinate grids.

- This scheme does not require a single transformation of the flow domain onto a square box. Thus, GIM routines can indeed treat arbitrary three-dimensional shapes.

- Grids generated for both internal and external flows in two and three dimensions have shown the versatility of the algebraic approach.

- The GIM code integration module has successfully computed flows on these complex grids, including the Space Shuttle main engine turbopump system.

- Plans for future application of the code include supersonic flow over missiles at angle of attack and three-dimensional, viscous, reacting flows in advanced aircraft engines. Plans for future grid generation work include schemes for time-varying networks which adapt themselves to the dynamics of the flow.

# BIBLIOGRAPHY

Spradley, L.W., J.F. Stalnaker and A.W. Ratliff, "Computation of Three-Dimensional Viscous Flows with the Navier-Stokes Equations," AIAA Paper 80-1348, July 1980.

Spradley, L.W., and M.L. Pearson, "GIM Code User's Manual for the STAR-100 Computer," NASA-CR-3157, Langley Research Center, Hampton, Va., 1979.

Spradley, L.W., P.G. Anderson and M. L. Pearson, "Computation of Three-Dimensional Nozzle-Exhaust Flows with the GIM Code," NASA CR-3042, Langley Research Center, Hampton, Va., August 1978.

Prozan, R.J., L.W. Spradley, P.G. Anderson and M.L. Pearson, "The General Interpolants Method," AIAA Paper 77-642, June 1977.

Gordon, W.J., and C.A. Hall, "Construction of Curvilinear Coordinate Systems and Applications to Mesh Generation," J. Numer. Math., Vol. 1, 1973, pp. 461-477.

# Component-Adaptive Grid Embeddng

E. H. Atta

Lockheed-Georgia Company

## Introduction:

One of the major problems related to transonic flow prediction about realistic aircraft configuration is the generation of a suitable grid which encompasses such configurations. In general, each aircraft component (wing, fuselage, nacelle) requires a grid system that is usually incompatible with the grid systems of the other components; thus, the implementation of finite-difference methods for such geometrically-complex configurations is a difficult task.

In this presentation a new approach is developed to treat such a problem. The basic idea is to generate different grid systems, each suited for a particular component. Thus, the flow field domain is divided into overlapping subdomains of different topology. These grid systems are then interfaced with each other in such a way that stability, convergence speed and accuracy are maintained.

## Model:

To evaluate the feasibility of the present approach a two-dimensional model is considered (figure 1). The model consists of a single airfoil embedded in rectangular boundaries, representing an airfoil in a wind tunnel or in free air. The flow field domain is divided into two overlapping subdomains, each covering only a part of the whole field. The inner subdomain employs a surface-fitted curvilinear grid generated by an elliptic grid-generator (ref. 1), while the outer subdomain employs a cartesian grid. The overlap region between the two subdomains is bounded by the outer boundry of the curvilinear grid and the inner boundary of the cartesian grid.

[1] Holst, T. L., "Implicit Algorithm for the Conservative Transonic Full-Potential Equation Using an Arbitrary Mesh," AIAA J., Vol. 17, No. 10, October 1979

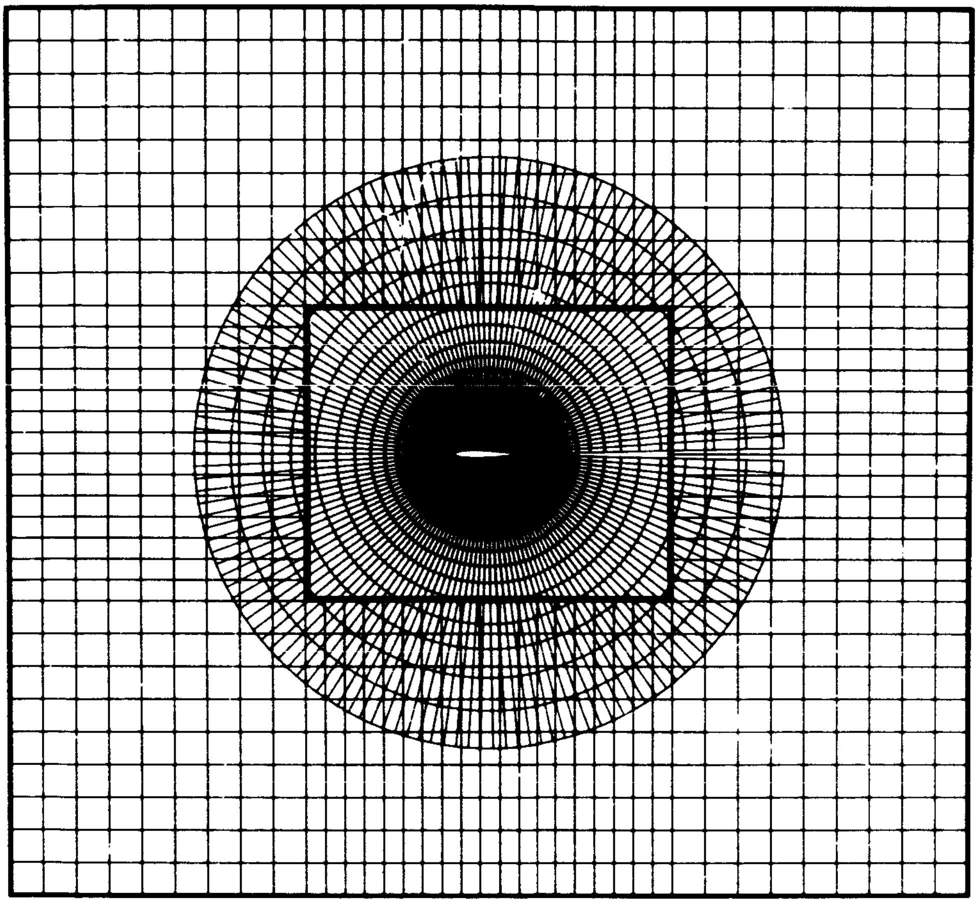


Figure 1. Composite grid for an airfoil

Approach:

The figure shows the two subdomains (A,B) of the flow field; each has a grid adapted to suit its geometry. The flow in both subdomains is governed by the transonic full-potential equation. While a Neuman-type boundary condition is used at the inner boundary of subdomain B (overlap inner boundary), a Dirichlet-type boundary condition is used at the outer boundary of subdomain A (overlap outer boundary). These boundary conditions are updated during the solution process. The implicit approximate factorization scheme is used in both grid systems. The code of ref. 1 is modified to fit into the present scheme.

The solution process is performed in cycles, starting by solving for the flow field in subdomain A, then switching after a number of iterations to solve for the flow field in subdomain B. During each cycle the overlap boundary conditions are updated by using two dimensional second order Lagrangian interpolation scheme. This process is then repeated until convergence is achieved in both subdomains.

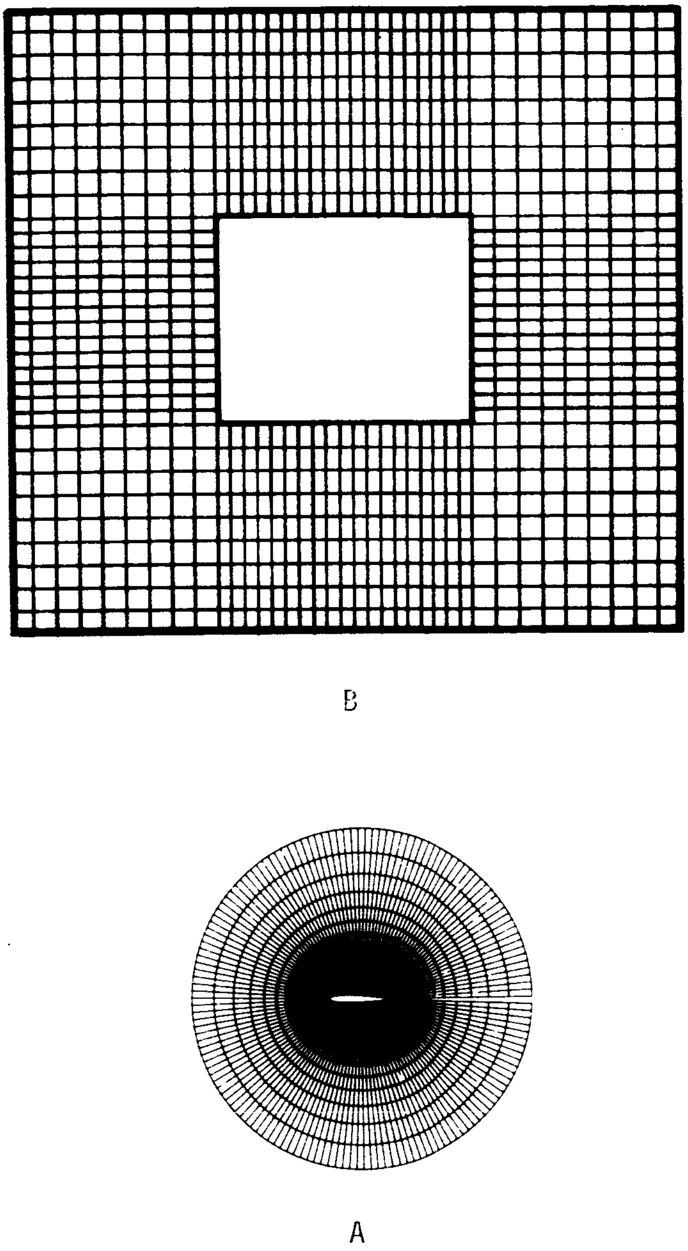


Figure 2. Grid topology for the different subdomains

Comparison with a homogeneous grid:

The results of the present method are compared with the results obtained from using one homogeneous grid for the entire flow field (ref. 1). In all the test cases considered, a standard grid with (31 x 147) points and a circular outer boundary located 6 chord-length away from the airfoil are used.

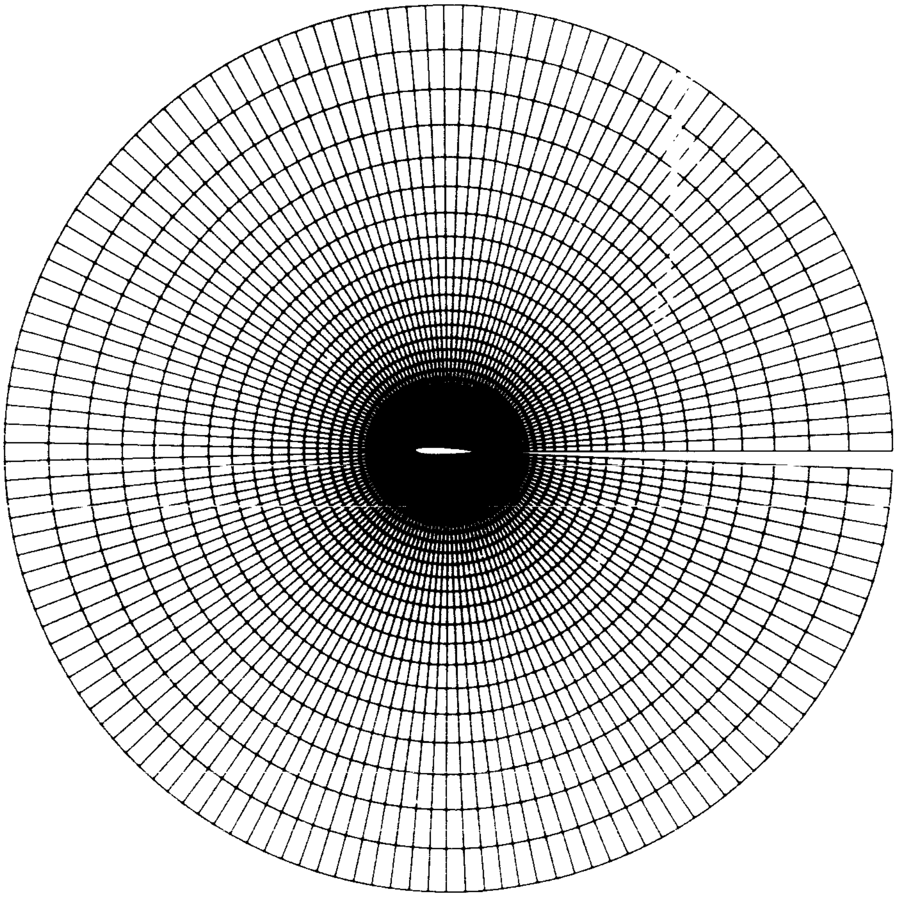


Figure 3. Uniform grid for an airfoil (ref. 1)

## Computed Results:

Results of the present method are compared with the results obtained from the code of ref. 1. Figures 4 and 5 display the pressure coefficient distribution for a NACA-0012 airfoil. The values of the different parameters affecting the performance of the numerical shceme are listed in table I. The results are in good agreement for both subcritical and supercritical cases; savings in computing time are achieved by reducing the size of the flow field covered by the curvilinear grid (subdomain A).

|  | Code of Ref. 1<br>TAIR Code | Present Method |  |
|---|---|---|---|
| Curvilinear grid | 31 x 147 | 15 x 147 | 21 x 147 |
| Cartesian grid |  | 30 x 30 | 30 x 30 |
| % cpu time reduction as compared to TAIR Code |  | 30% | 10% |
| location of subdomain B outer boundary |  | 6 chord-length | 6 chord-length |
| Location of subdomain B inner boundary |  | 1 chord-length | 2 chord-length |
| Location of subdomain A outer boundary |  | 1 chord-length | 4 chord-length |
| number of cycles for convergence |  | 9 | 10 |

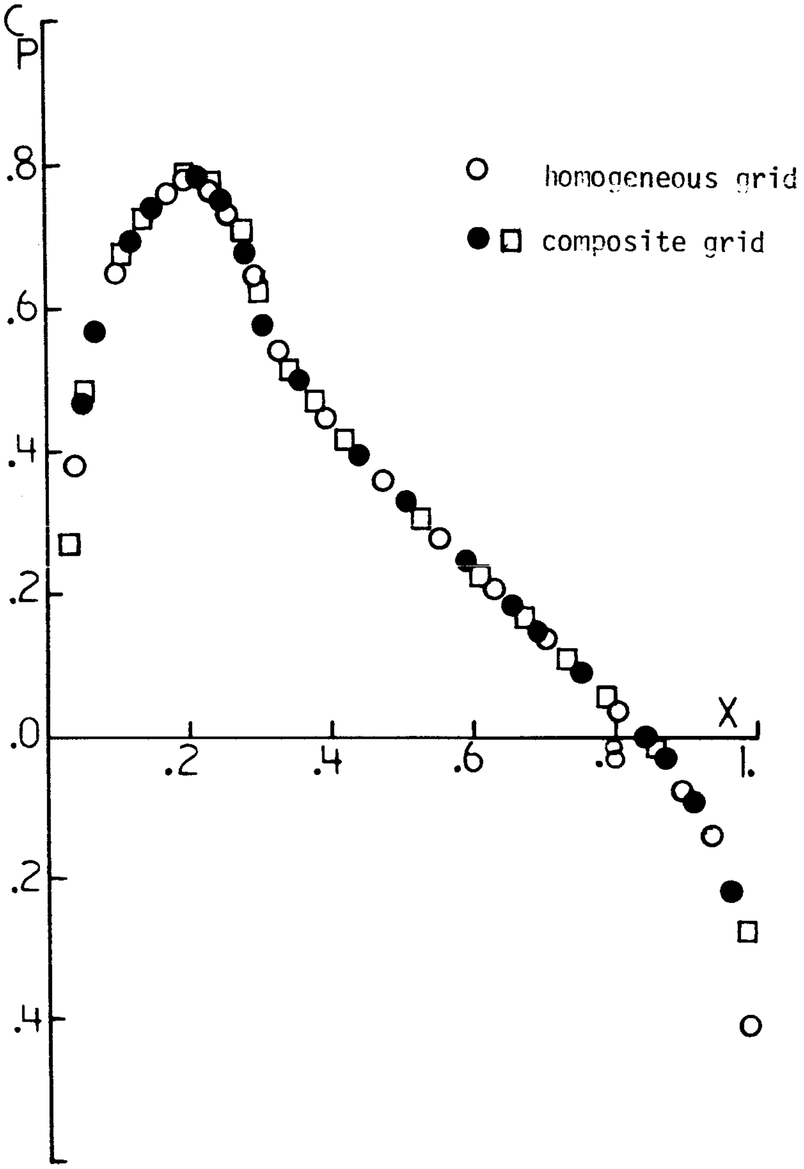


Figure 4. Comparison of pressure coefficient for NACA-0012 (M∞ = .75, α = 0.)

| | Code of Ref. 1<br>TAIR Code | Present Method | |
|---|---|---|---|
| Curvilinear grid | 31 x 147 | 18 x 147 | 14 x 147 |
| Cartesian grid | | 30 x 30 | 50 x 50 |
| % cpu time reduction as compared to TAIR Code | | 20% | 10% |
| location of subdomain B outer boundary | | 6 chord-length | 6 chord-length |
| location of subdomain B inner boundary | | 1 chord-length | 1/4 chord-length |
| location in subdomain A outer boundary | | 2 chord-length | 1 chord length |
| number of cycles for convergence | | 12 | 15 |

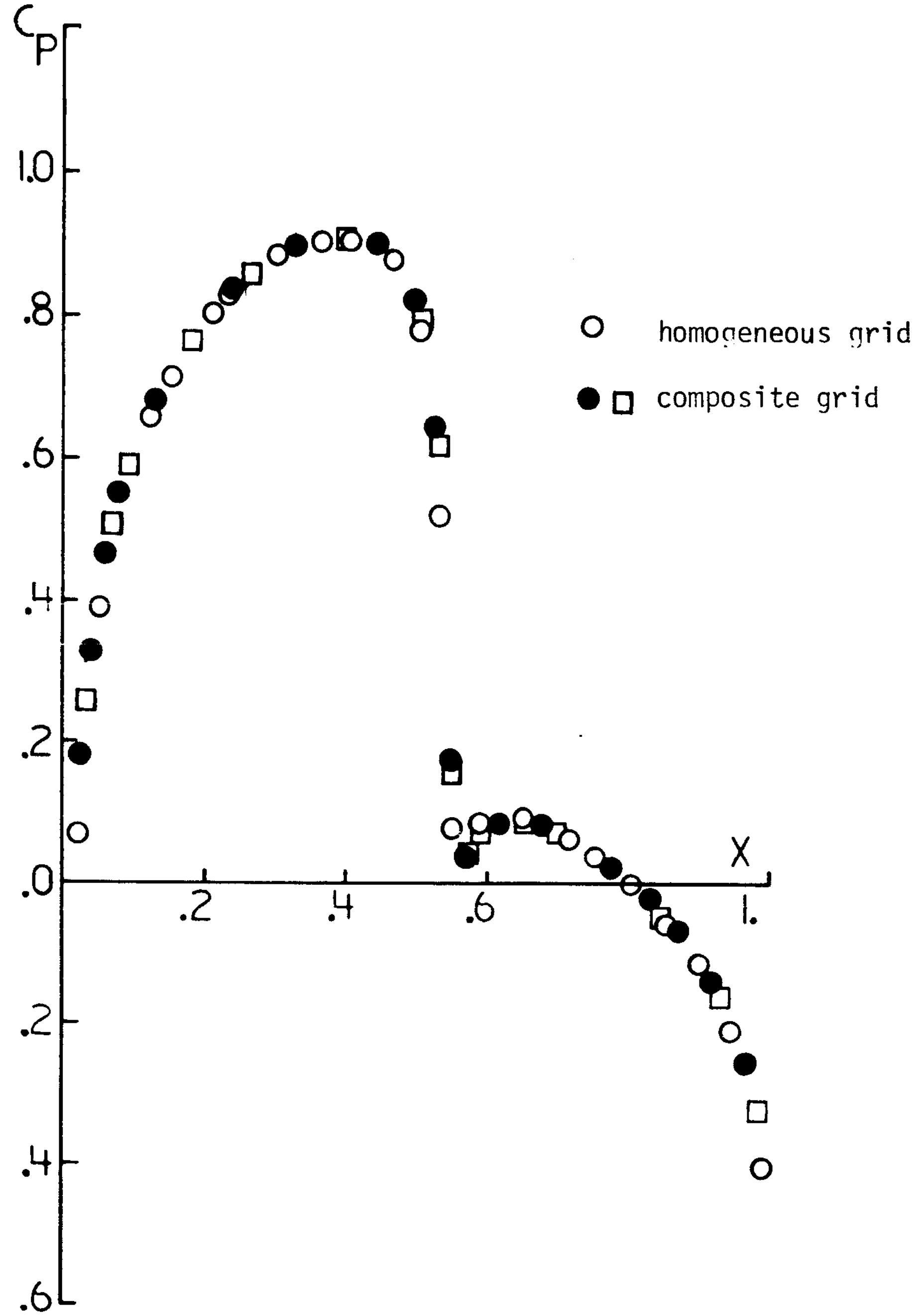


Figure 5. Comparison of pressure coefficient for NACA-0012 ($M_\infty = .8$, $\alpha = 0.$)

## Flow Field Topology :

The extent of the overlap region between the different grids and the relative size of each subdomain are the main factors affecting the accuracy and convergence speed of the present scheme. The figure shows the flow field topology for several test cases. In these cases the overlap extent and subdomain sizes are varied to determine their optimum values that will minimize the computing effort, while maintaining a reasonable accuracy.

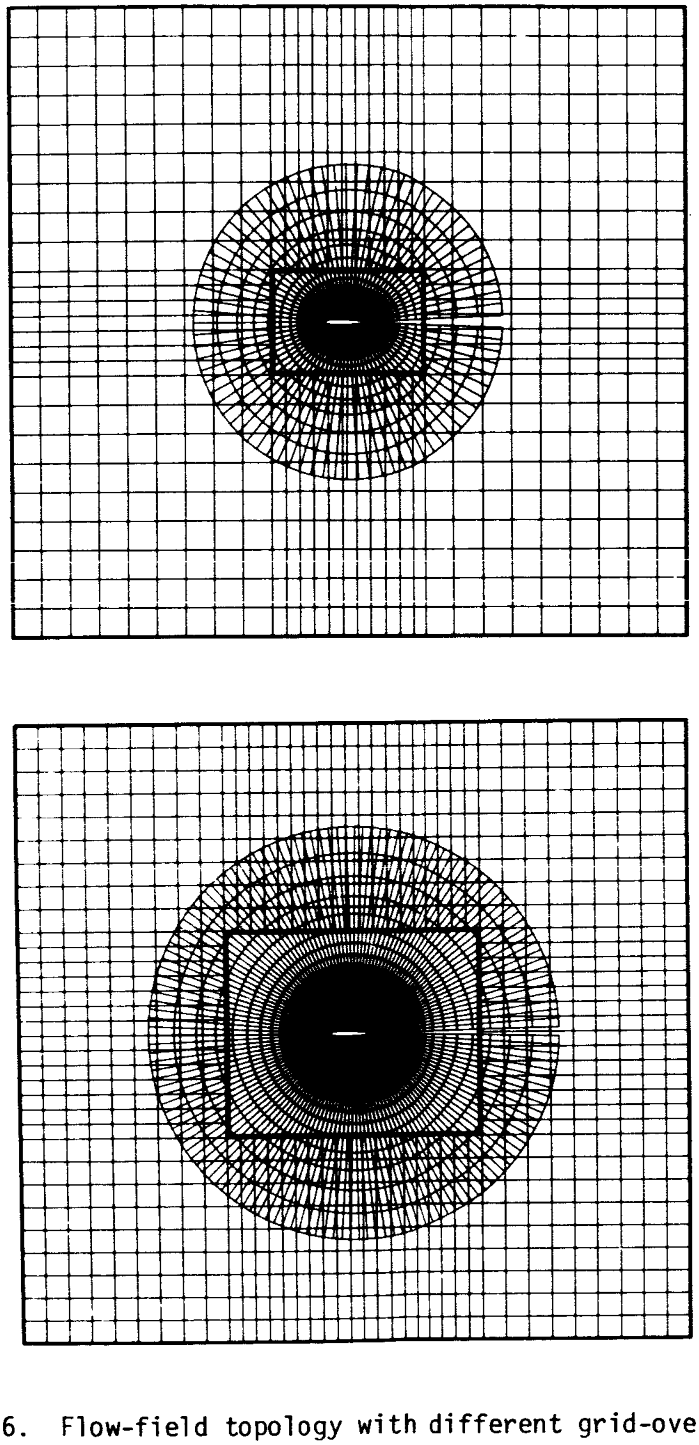


Figure 6. Flow-field topology with different grid-overlap

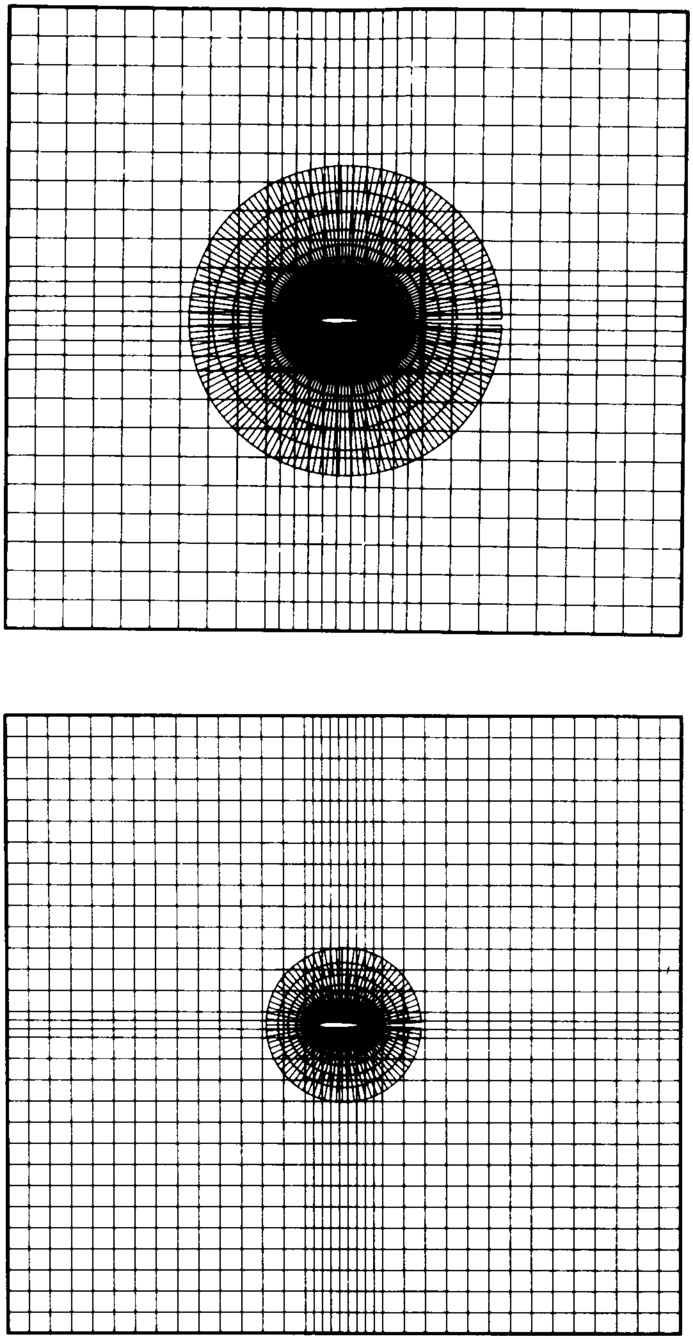


Figure 6. Flow-field topology with different grid-overlap (cont'd)

Overlap arrangement:

Test Cases with different grids-arrangement are compared to determine the optimum choice for the extent of the overlap region. A work factor w [number of iterations for convergence x number of grid points (curvilinear grid)] is taken as a measure of the computing effort. Numerical results show that increasing the extent of the overlap region decreases the number of iterations for convergence, however, this also increases the computing effort. To minimize the computing time the cartesian grid should overlap 15-25% of the curvilinear grid, also the inner boundary of the cartesian grid should not be located less than 0.25 chord-length away from the airfoil.

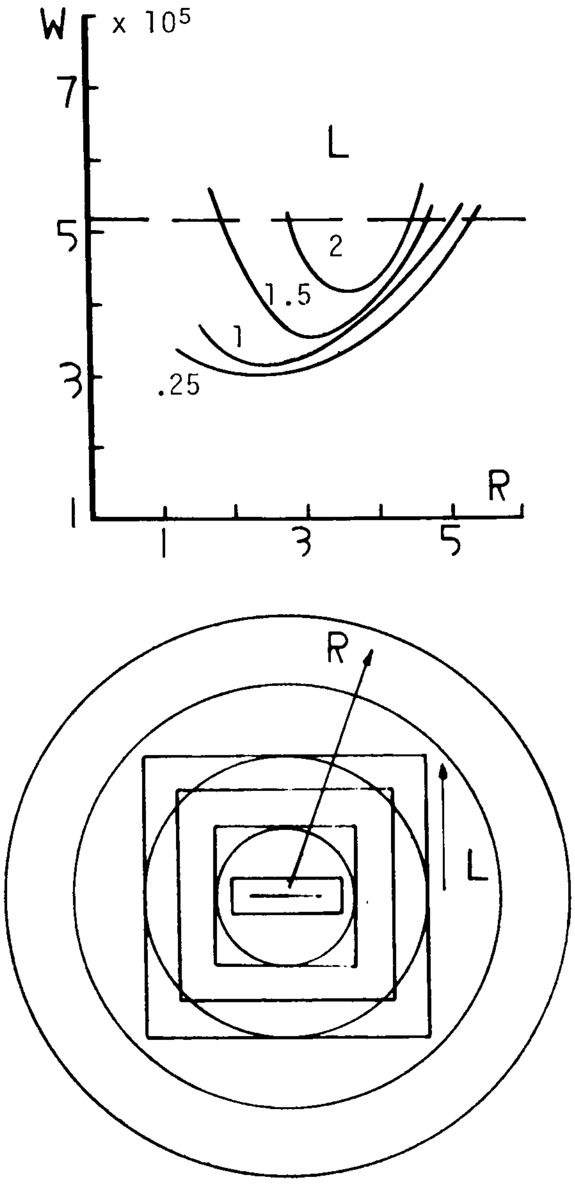


Figure 7. Effect of overlap parameters on work factor w (NACA-0012, $M_\infty$ = .8, $\alpha$ = 0.)

Computed Results:

The use of nonoptimal parameters for grids arrangement (overlap extent, relative grid sizes) can produce inaccurate results and/or slow down convergence. The Peaky pressure coefficient distribution shown in the figure is corrected by increasing the extent of the overlap region

| | Code of Ref. 1<br>TAIR Code | Present Method | |
|---|---|---|---|
| Curvilinear grid | 31 x 147 | 10 x 147 | 15 x 147 |
| Cartesian grid | | 30 x 30 | 40 x 40 |
| location of subdomain B outer boundary | | 6 chord-length | 6 chord-length |
| location of subdomain B inner boundary | | 1/4 chord- | 1/4 chord- |
| location of subdomain A outer boundary | | 1.5 chord- | 1 chord- |

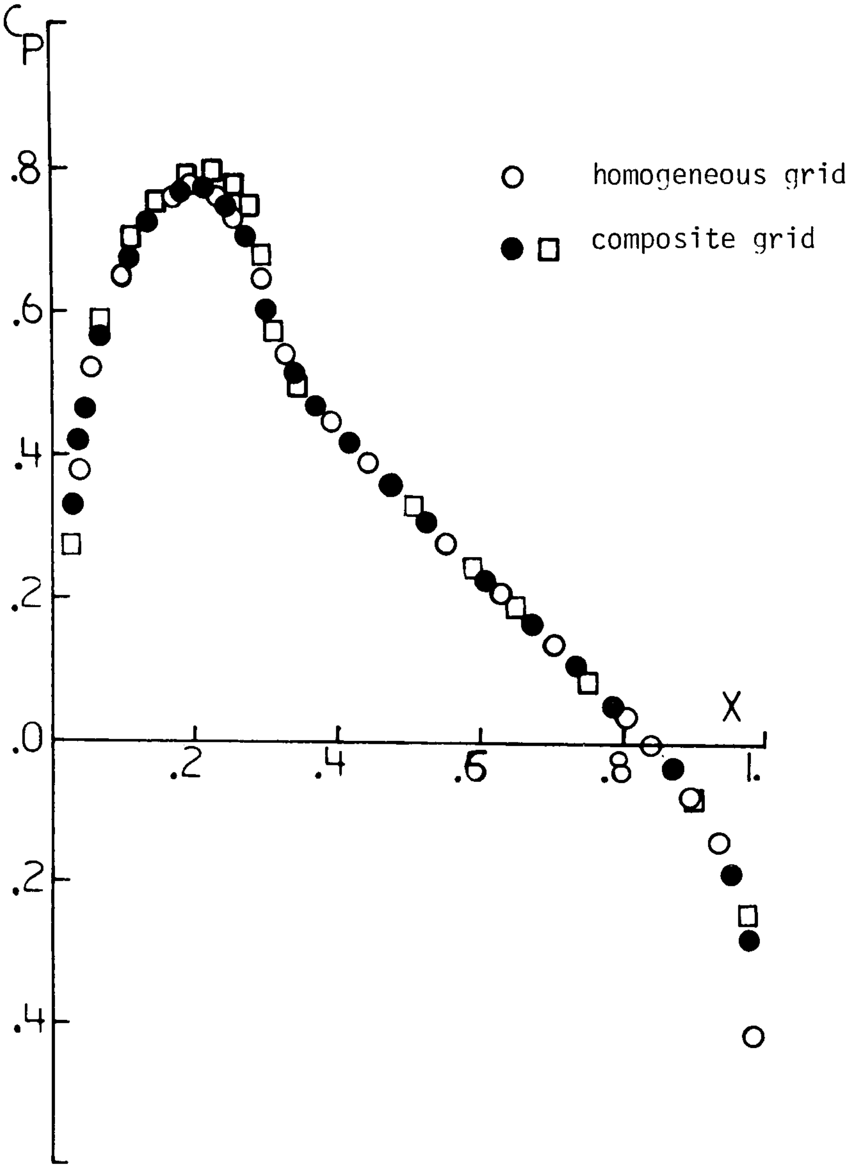


Figure 8. Comparison of pressure coefficient for NACA -0012 ($M_\infty$ = .75, $\alpha$ = 0.)

Computed Results:

Figures 9 and 10 display the pressure coefficient distribution for two lifting cases. The evolution of circulation, and hence lift, is slowed down as the solution process alternates between the different grids. This dealt with by decreasing the number of iterations performed in each grid.

| | Code of Ref. 1<br>TAIR Code | Present Method |
|---|---|---|
| Curvilinear grid | 31 x 147 | 15 x 147 |
| Cartesian grid | | 30 x 40 |
| % cpu time reduction as compared to TAIR Code | | 39% |
| location of subdomain B outer boundary | | 6 chord-length |
| location of subdomain B inner boundary | | 1 chord-length |
| location of subdomain A outer boundary | | 3 chord-length |
| lift coefficient | 0.334 | 0.337 |
| number of cycles for convergence | | 16 |

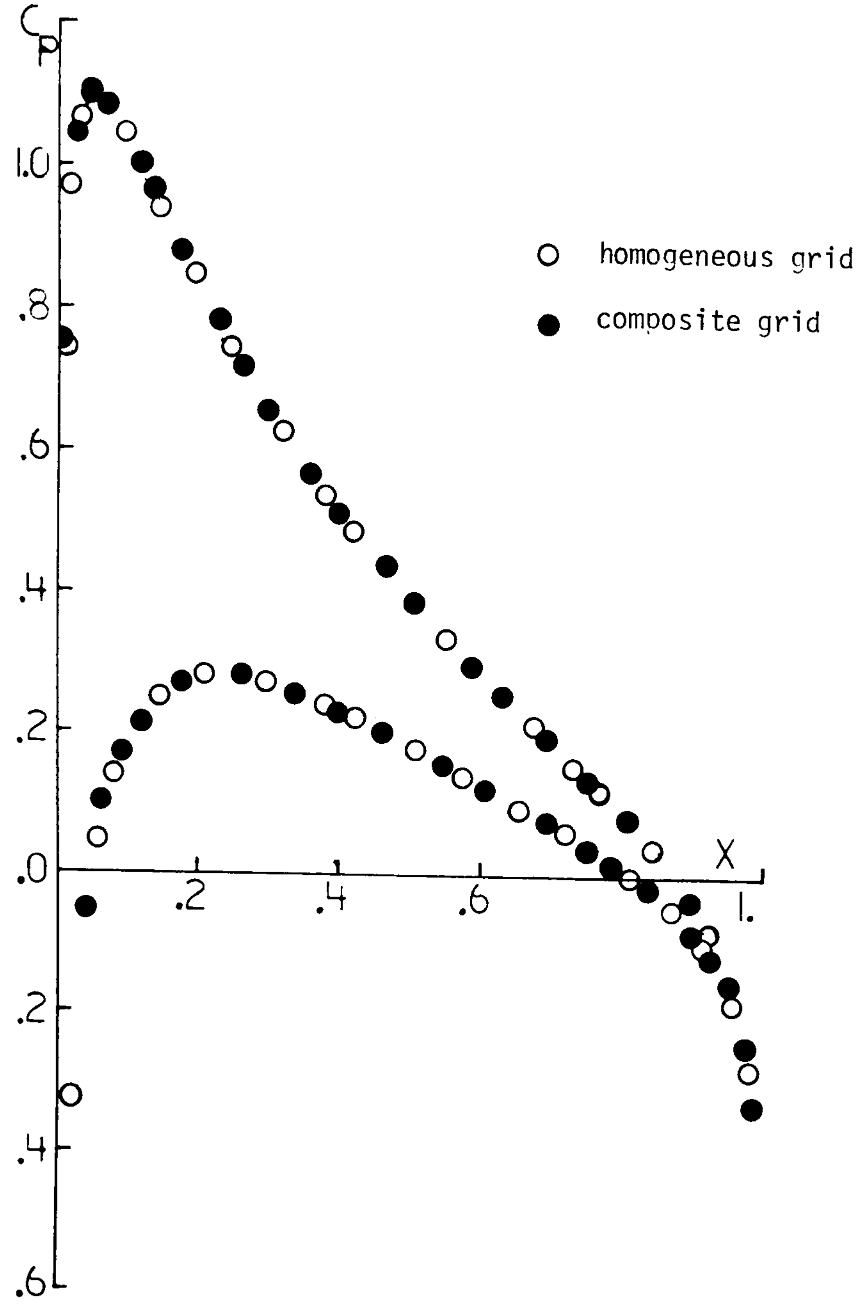


Figure 9. Comparison of pressure coefficient for NACA-0012 ($M_\infty$ = .63, $\alpha$ = 2°)

|  | Code of Ref. 1<br>TAIR Code | Present Method |
|---|---|---|
| Curvilinear grid | 31 x 147 | 21 x 147 |
| Cartesian Grid |  | 30 x 30 |
| % cpu time reduction as compared to TAIR Code |  | 2% |
| location of subdomain B outer boundary |  | 6 chord-length |
| location of subdomain B inner boundary |  | 2 chord-length |
| location of subdomain A outer boundary |  | 4 chord-length |
| lift coefficient | 0.574 | .584 |
| number of cycles for convergence |  | 14 |

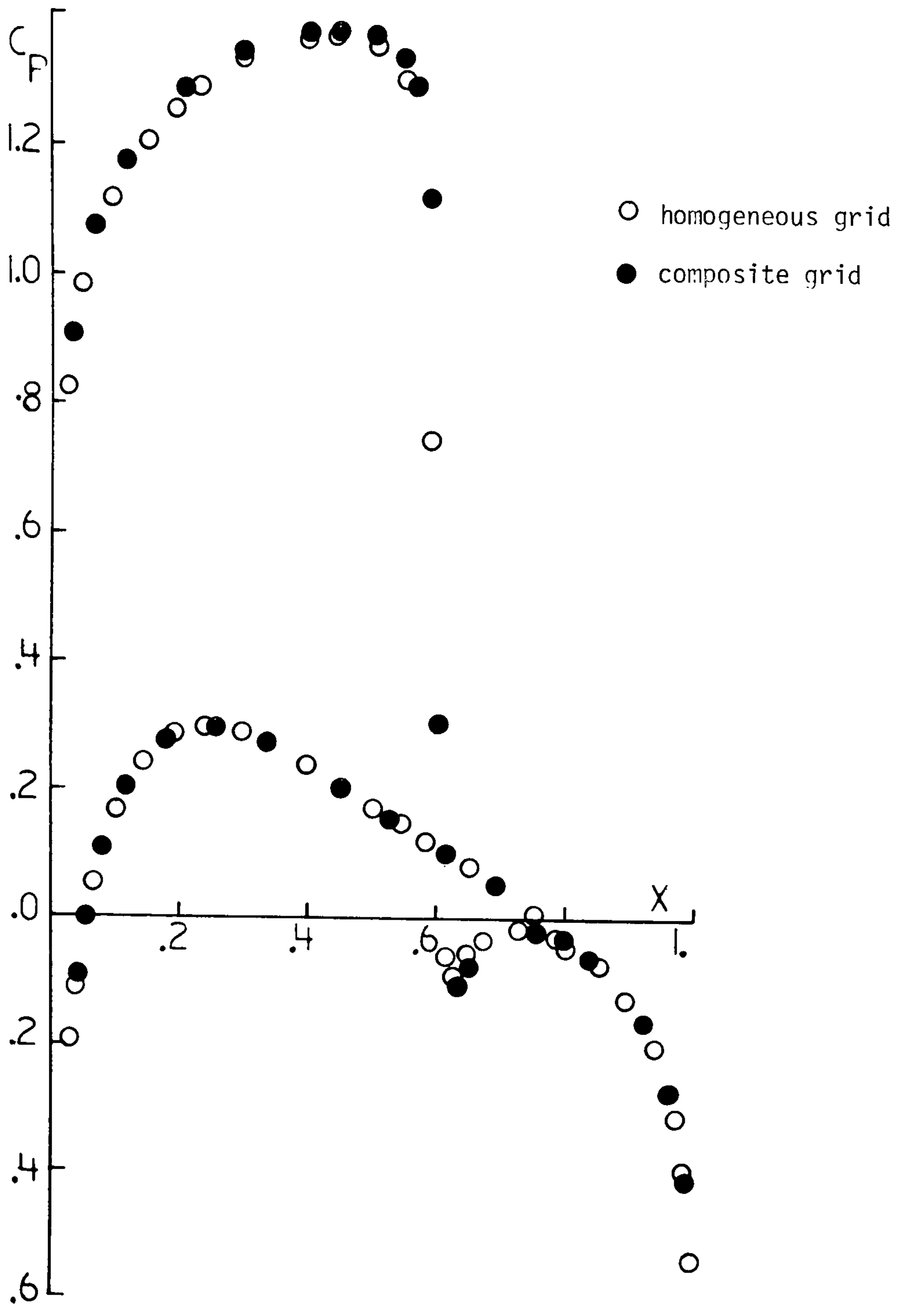


Figure 10. Comparison of pressure coefficient for NACA-0012 ($M_\infty$ = .75, $\alpha$ = 2°)

Errors in Sonic line position :

Should the shock wave extend into the overlap region, the interpolation process can produce errors in the shock location and strength. Comparisons of the results of the present method with those of a homogeneous grid shows that the maximum relative error did not exceed 1.5%.

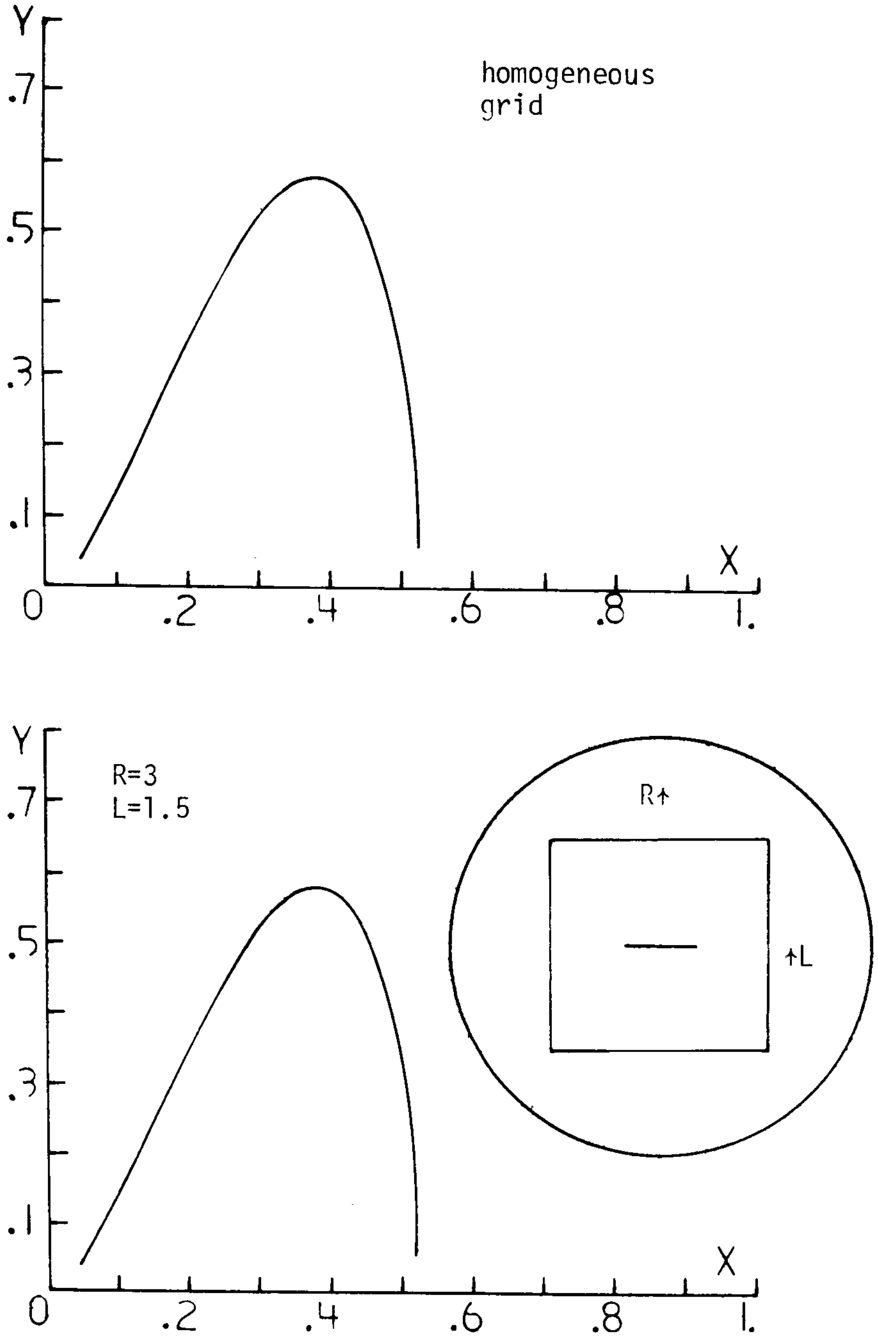


Figure 11. Effect of interface location on sonic line position (NACA-0012, $M_\infty$ = .8, $\alpha$ = 0.)

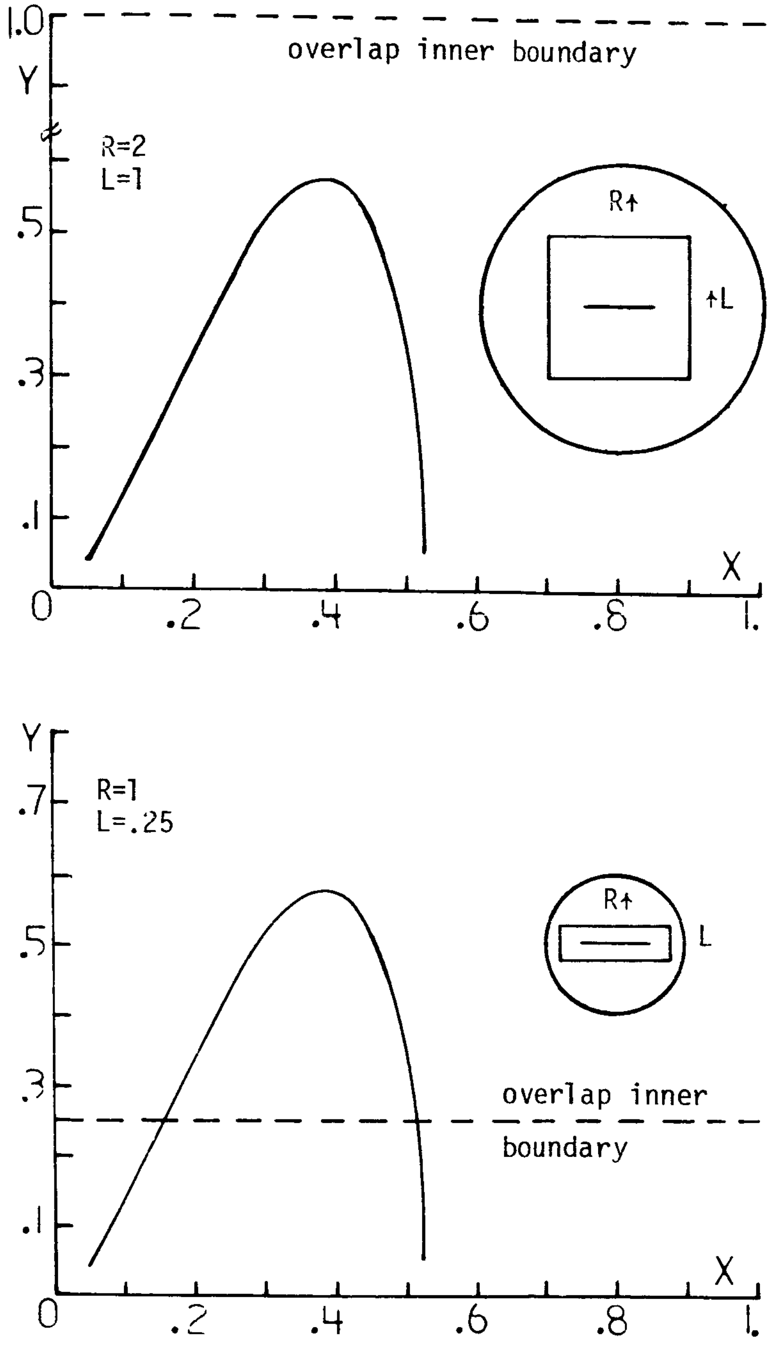


Figure 11. Effect of interface location on sonic line position (NACA-0012, M∞ = .8, α = 0.) (cont'd)

## Conclusion:

A method for interfacing grid systems of different topology is developed. This offers a new approach to the problem of transonic flow prediction about multiple-component configurations. The method is implemented in a 2-D domain containing two grid systems of different topology. The numerical scheme in the present method proved to be stable and accurate. Savings in computer time and/or storage is achieved by the proper choice of the overlap region between the different grids.

# A Simple Numerical Orthogonal Coordinate Generator for Fluid Dynamic Applications

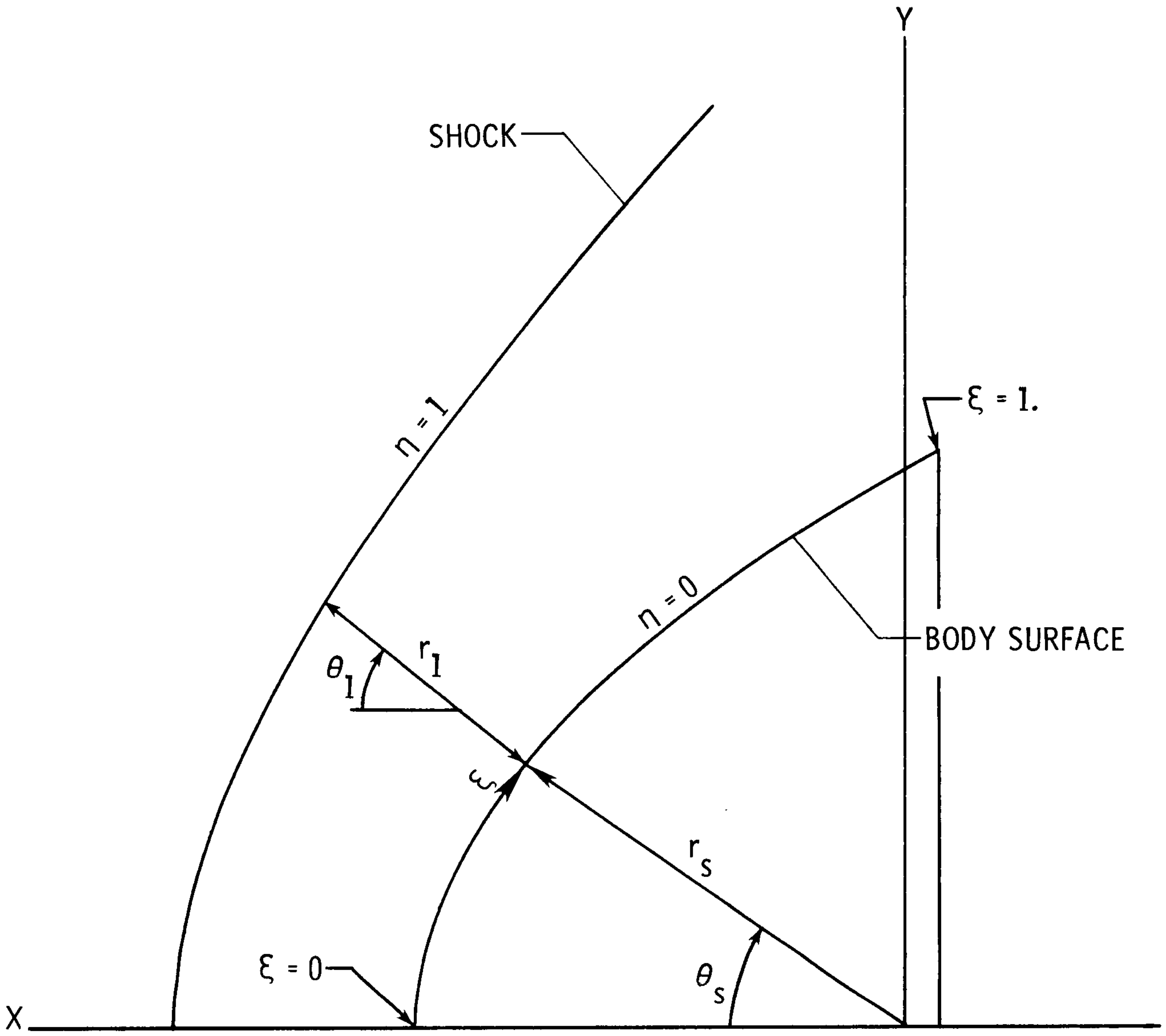
by

Randolph A. Graves, Jr.

OAST Aerodynamics Office
NASA Headquarters
Washington, DC

# Abstract

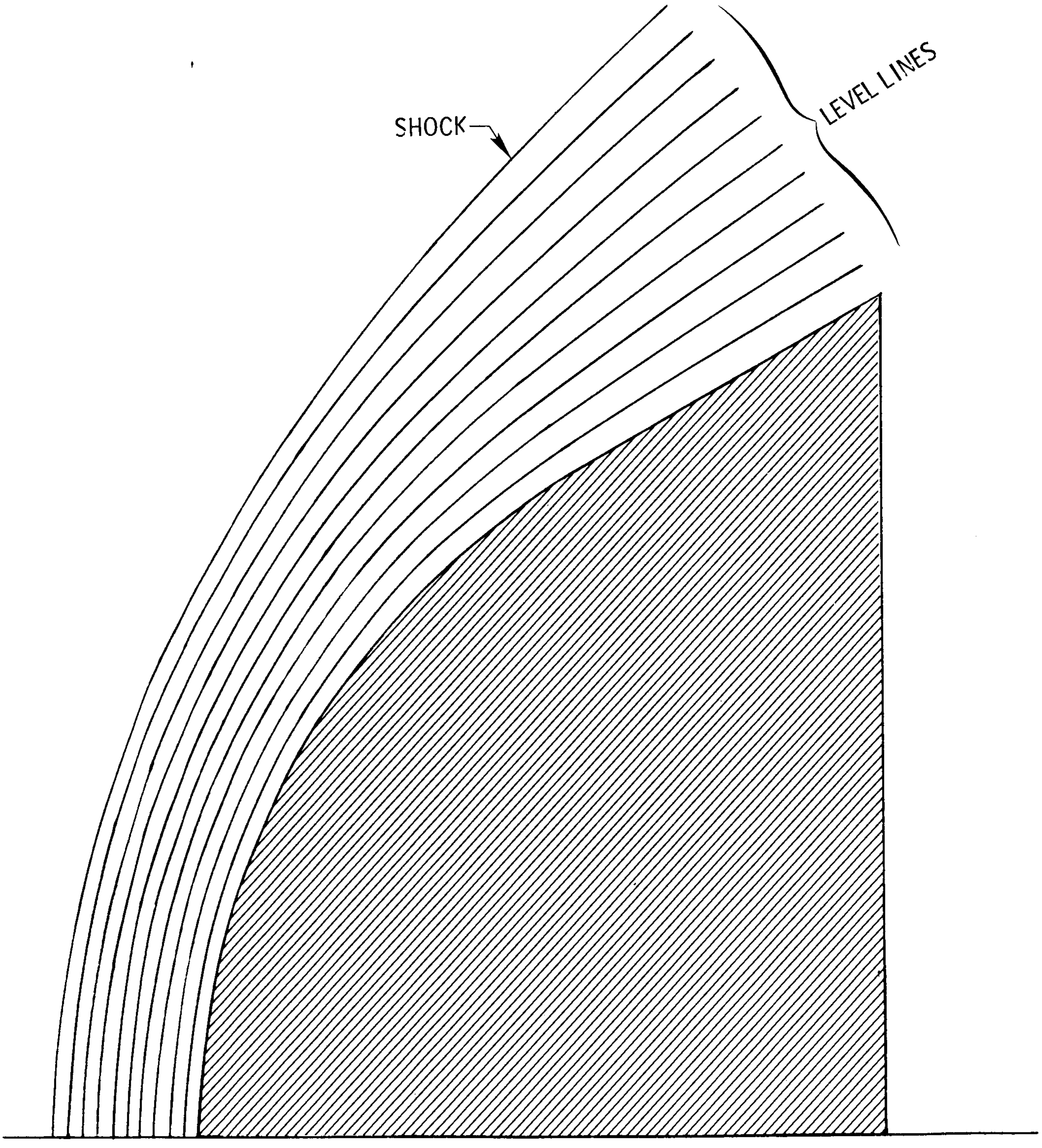
An application of a simple numerical technique which allows for the rapid construction of orthogonal coordinate systems about two dimensional and axisymmetric bodies is presented. This technique which is based on a "predictor-corrector" numerical method is both simple in concept and easy to program. It can be used to generate orthogonal meshes which have unequally spaced points in two directions. These orthogonal meshes in their transformed computational plane are, however, equally spaced so that the differencing for the metric coefficients and the fluid dynamic equation terms can be easily determined using equally spaced central finite differences. Solutions to the Navier Stokes equations for flow over blunt bodies with reverse curvature are presented. The coupling of the time dependent fluid dynamic equations and the coordinate generator worked well with no undersirable effects noted.



Flowfield geometrical relationships.

## Flowfield Geometrical Relationships

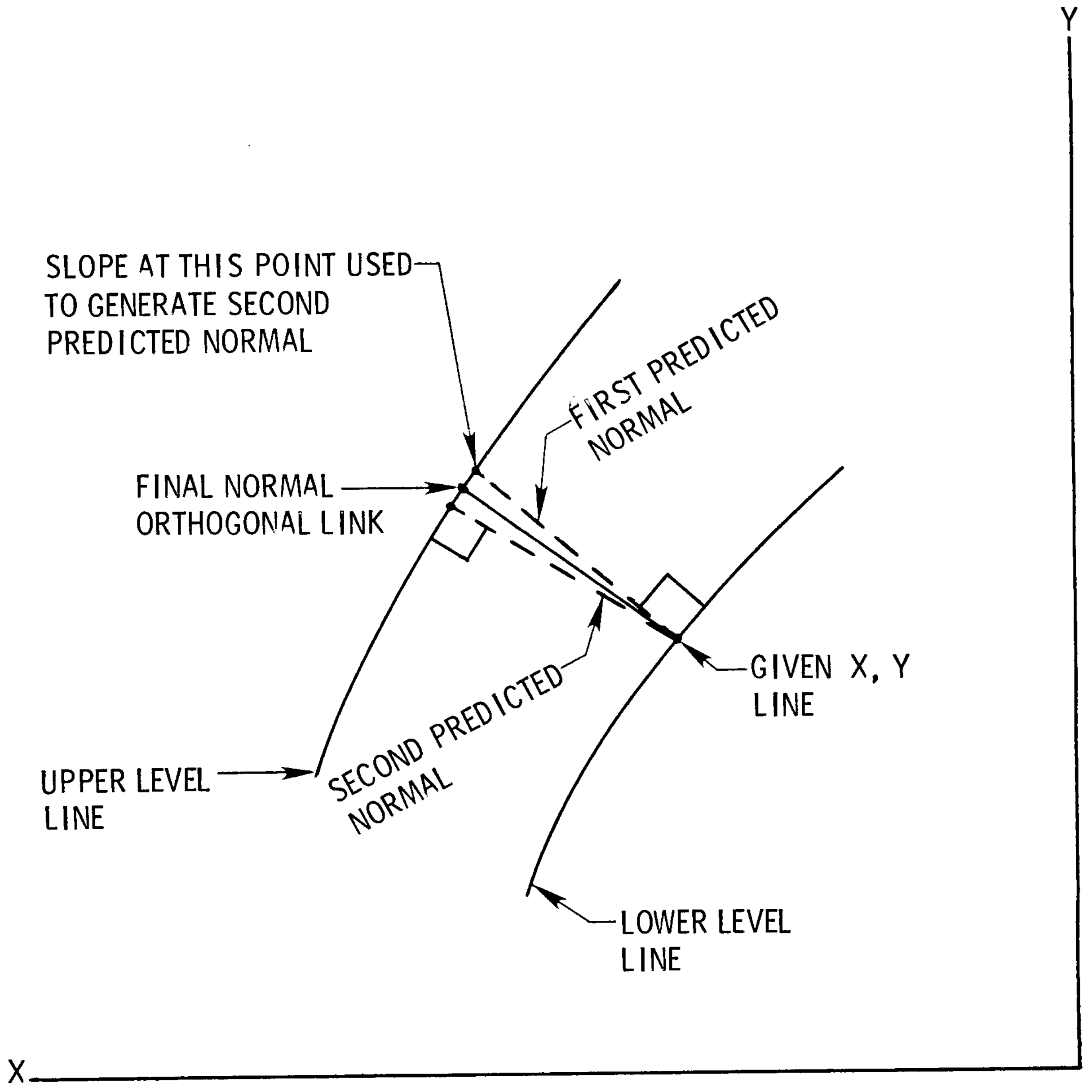
The numerically generated orthogonal coordinates are determined from the original cartesian coordinate systems description of the body surface and outer boundary. Taking the origin of the X,Y system as lying inside the body to be described, the surface distance $\xi$, which forms one of the transformed orthogonal coordinates, can be easily calculated by defining $\xi$ as zero at origin of the region of interest and increasing to unity at the end of the region (nondimensionalized surface distance). The other orthogonal coordinate, $\eta$, is taken as zero on the body surface and as unity on the outer boundary. Thus the region of interest is transformed into a nondimensional square.



Level line construction.

## Level Line Construction

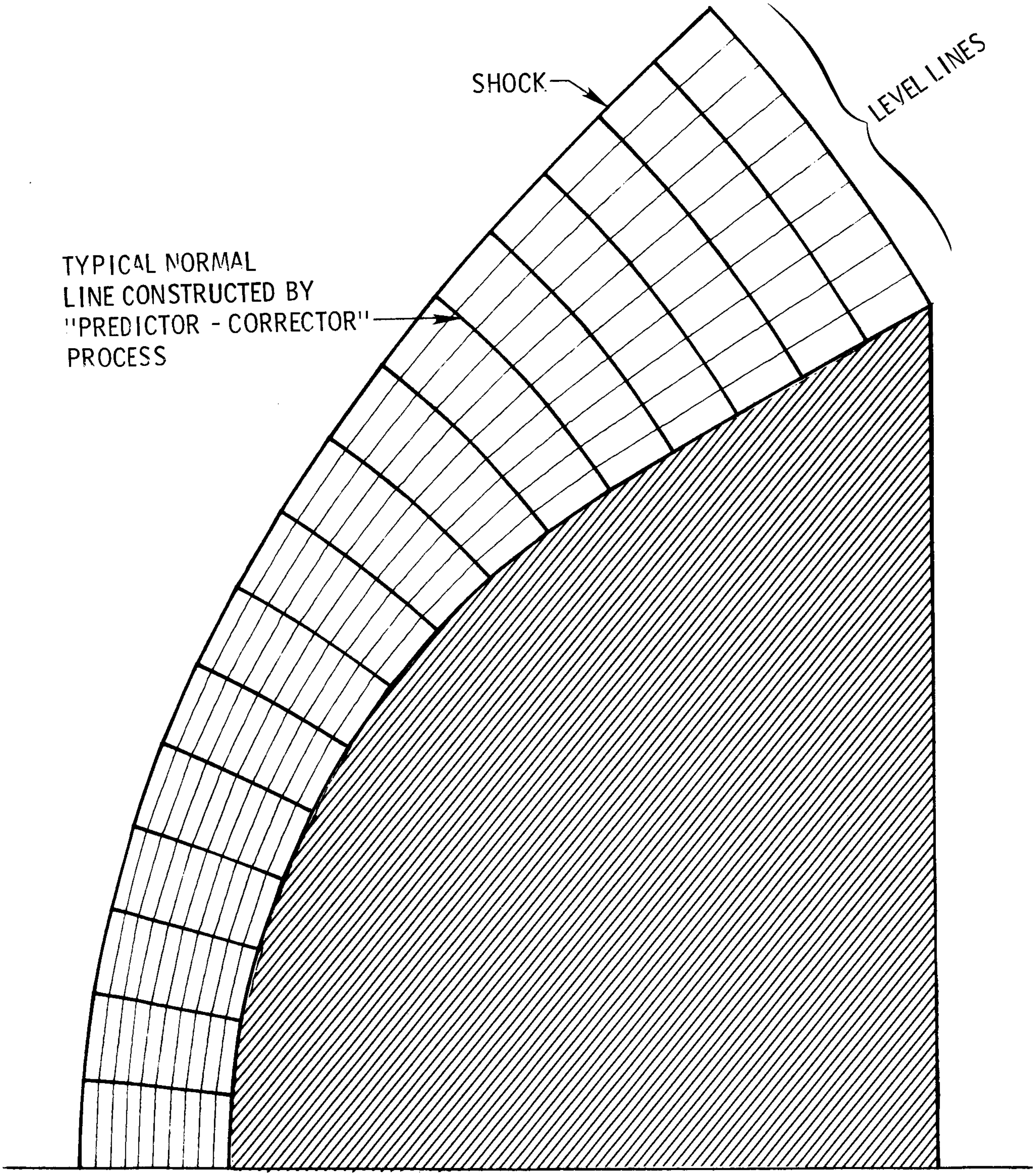
The level lines between the outer boundary and the body surface can be constructed arbitrarily, however, the easiest approach is to construct the level lines along straight lines connecting corresponding points on the body and the outer boundary. The mesh points on the outer boundary are not the final mesh points but initial values used only to set up the level lines. The actual mesh points will result from the numerical generation of the orthogonal normal lines. The spacing of the level lines is arbitrary and highly stretched meshes can be easily constructed.



Normal line construction technique.

## Normal Line Construction Technique

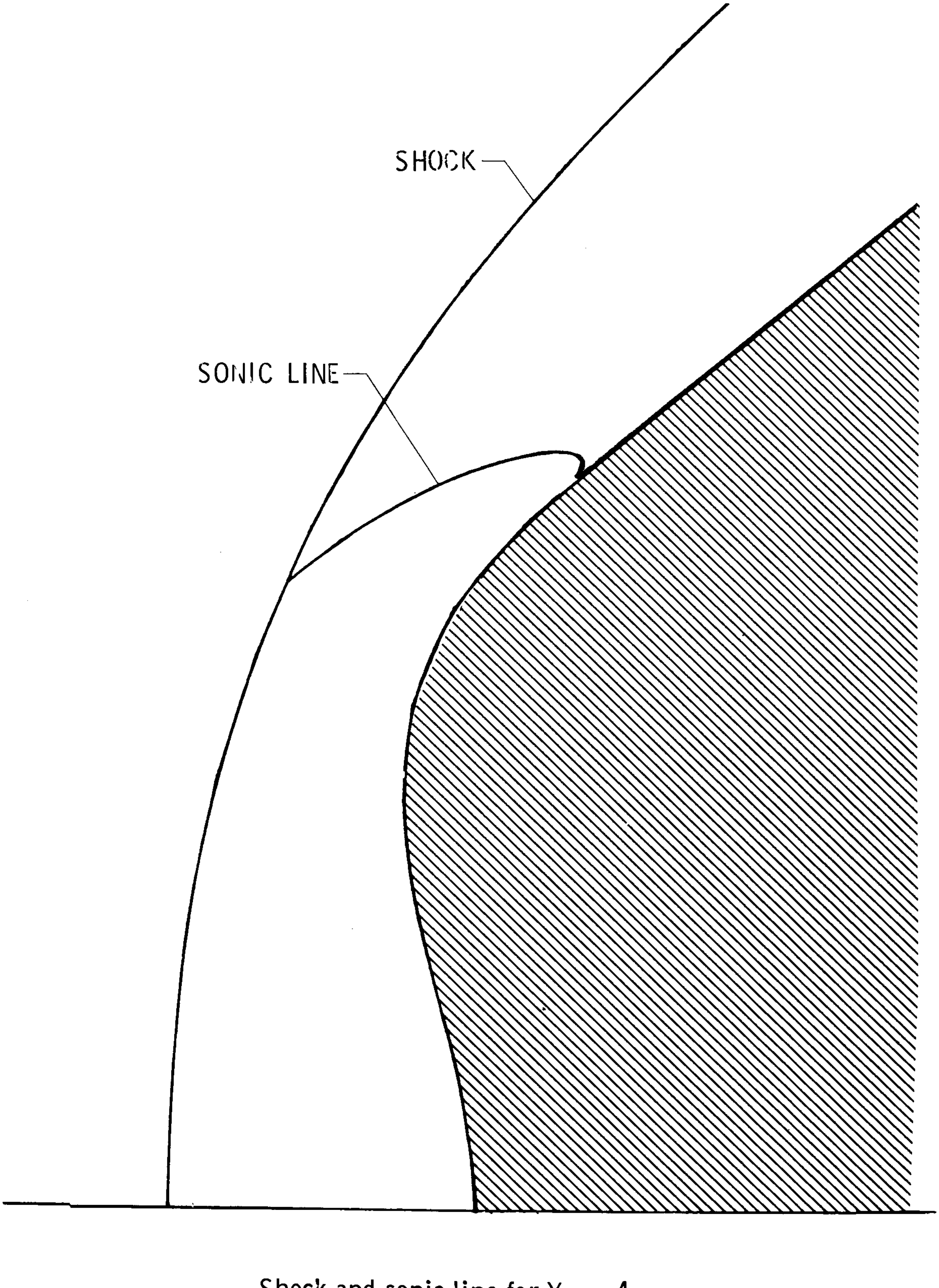
Once the level lines have been determined, the normal lines are constructed numerically so that an orthogonal system is defined. The approach to the construction of the normal lines is the one given by McNally which uses a simple "predictor-corrector" technique analogous to the trapezoidal integration method of numerical integration. In this technique, the solution is first predicted from the level line at a known point by using the Euler method. Once the predicted point on the next level line is obtained, the slope at that point is calculated and a new predicted point is obtained using this slope. The actual solution is then a combination of these two solutions, i.e. the final X,Y values are an average of the predicted and corrected ones.


SHOCK
LEVEL LINES
TYPICAL NORMAL LINE CONSTRUCTED BY "PREDICTOR - CORRECTOR" PROCESS

Typical coordinate mesh construction.

## Typical Coordinate Mesh Construction

Starting on the body, the normal line construction technique proceeds point by point along a level line until all normals on that level have been constructed. The solution then proceeds to the next level and the process is continued until the outer boundary is reached. Thus the complete mesh system is numerically generated in a simple straight forward, non iterative, process. Since the computational plane $(\xi,\eta)$ is an equally spaced rectangular region, the metric coefficients can be determined from the completed mesh system using equally spaced finite difference relations. Fourth order accurate difference relations are recommended as they provide for smoothly varying metric coefficients.



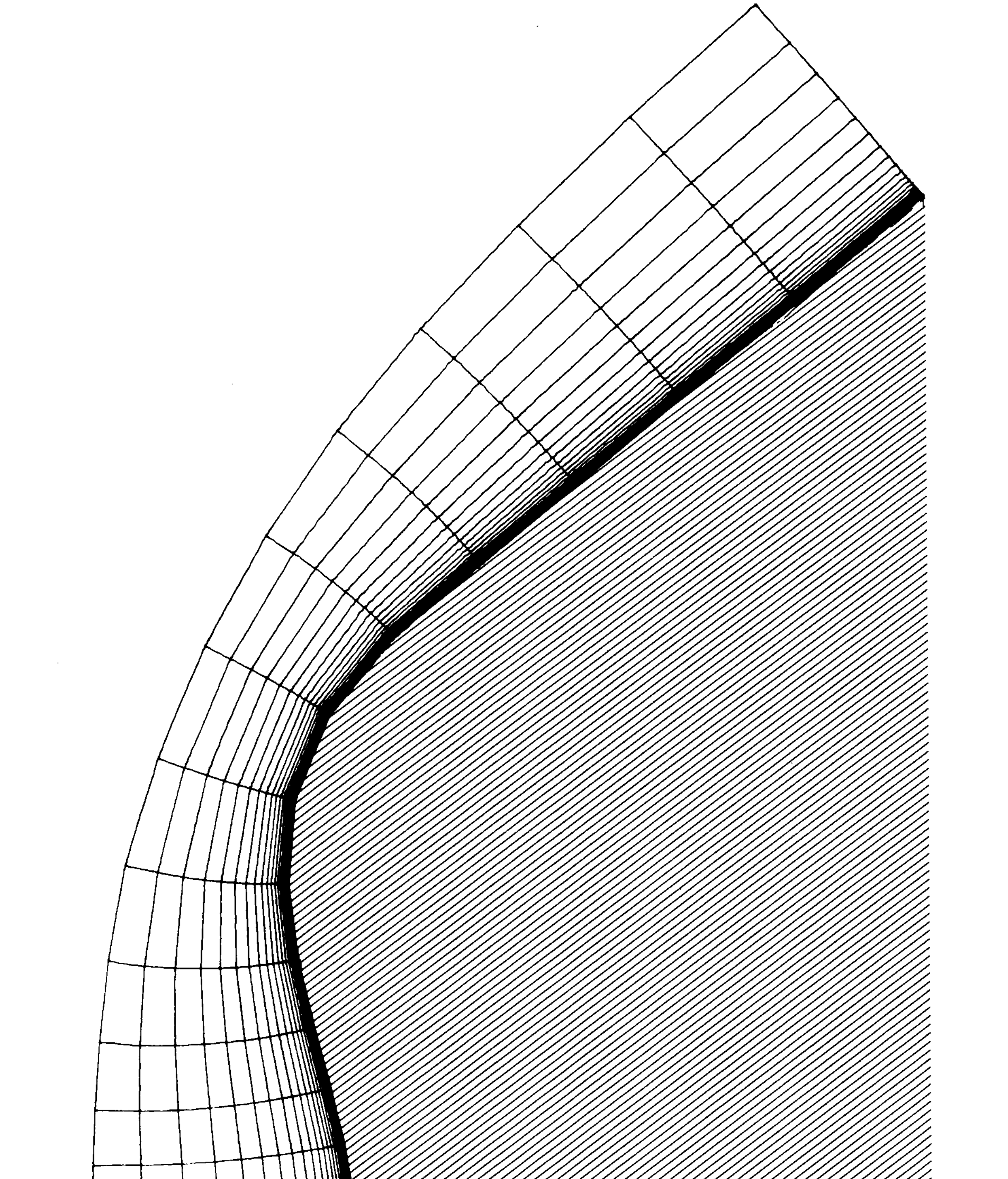
Shock and sonic line for $X_0 = .4$.

## Shock and Sonic Line

Solutions to the laminar flow Navier-Stokes equations were obtained for flow over bodies with blunted noses, including reverse curvature. These bodies were generated using the following cubic forebody generator,

$$X=X_o +A_1 y^2 +A_2 y^3$$

where $X_o$determines the nose offset while the coefficients $A_1$ and $A_2$ are determined such that the forebody nose section joins smoothly to the conical flank. This solution was run for a free stream Mach number of 10.33 and $X_o$=.4. The shock shape and sonic line are typical of the solution for bodies with very blunt nose regions.



Converged coordinate system for $X_0 = .4$.

## Converged Coordinate System

The converged coordinate system shown for $X_O$=.4 is composed of 15 transverse stations and 31 normal stations. The normal direction spacing is highly stretched to provide resolution for the boundary layer. There is only mild stretching in the transverse direction to provide for improve stagnation region resolution. There were no undesirable effects noted in the coupling of the viscous flow calculations with the coordinate generation.

# A THREE-DIMENSIONAL BODY-FITTED COORDINATE SYSTEM FOR FLOW FIELD CALCULATIONS ON ASYMMETRIC NOSETIPS

DARRYL W. HALL

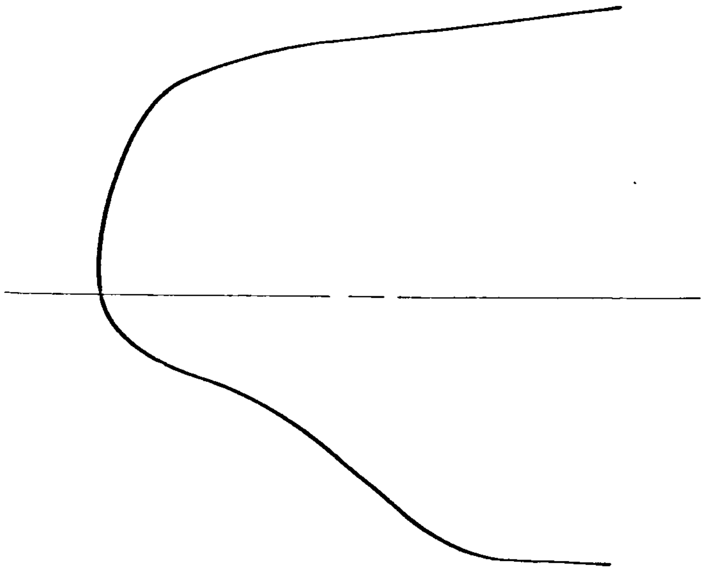
SCIENCE APPLICATIONS, INC.
COMPUTATIONAL FLUID DYNAMICS DIVISION
(215) 687-5080

# ABSTRACT

This presentation describes a three-dimensional body-fitted coordinate system developed for use in the calculation of inviscid flows over ablated, asymmetric reentry vehicle nosetips. Because of the potential geometric asymmetries, no standard coordinate system (e.g., spherical, axisymmetric reference surface-normal) is capable of being closely aligned with the nosetip surface. To generate a 3-D, body-fitted coordinate system an analytic mapping procedure is applied that is conformal within each meridional plane of the nosetip; these transformations are then coupled circumferentially to yield a three-dimensional coordinate system. The mappings used are defined in terms of "hinge points", which are points selected to approximate the body contours in each meridional plane. The selection of appropriate hinge points has been automated to facilitate the use of the resulting nosetip flow field code.

## PROBLEM DEFINITION

CALCULATION OF SUPERSONIC/HYPERSONIC INVISCID FLOWS OVER ASYMMETRIC ABLATED REENTRY VEHICLE NOSETIPS



ASYMMETRIC ABLATED NOSETIP SHAPE

## APPROACH

- FINITE-DIFFERENCE SOLUTION OF UNSTEADY EULER EQUATIONS
- STEADY FLOW SOLUTION SOUGHT AS THE ASYMPTOTIC LIMIT OF UNSTEADY FLOW

## PROBLEM DEFINITION

The goal of this effort is the development of a procedure for calculating supersonic/hypersonic inviscid flows over asymmetric ablated reentry vehicle nosetips. These asymmetric shapes, such as illustrated in this figure, result from asymmetric transition on the nosetip, which occurs at the lower altitudes during reentry (i.e., below 50 KFT). Because these shapes occur in the high Reynolds number, turbulent regime, with thin boundary layers, an inviscid solution is capable of accurately predicting the pressure forces on the nosetip. The nosetip flow field solution is also required to provide the required initial data for afterbody calculations; this coupling of nosetip and afterbody codes allows accurate prediction of the effects of the nosetip shape on the afterbody flow field.

The flow field code developed is a finite-difference solution of the unsteady Euler equations in "non-conservation" form (i.e., the dependent variables are the logarithm of pressure, P, the velocity components, u,v,w, and the entropy, s). In this approach the steady flow solution is sought as the asymptotic limit of an unsteady flow, starting from an assumed initial flow field.

## COORDINATE SYSTEM REQUIREMENTS

OPTIMUM COORDINATE SYSTEM FOR NUMERICAL FLOW FIELD CALCULATIONS IS BODY-ORIENTED

COORDINATE TRANSFORMATION SOUGHT THAT:

1.) ALIGNS COORDINATE SURFACES WITH BODY SURFACE

2.) IS ANALYTIC (SOLUTION OF PDE'S NOT REQUIRED TO DEFINE TRANSFORMATION)

3.) CAN BE READILY AUTOMATED (TO MINIMIZE INPUTS REQUIRED FROM USER)

## COORDINATE SYSTEM REQUIREMENTS

It is well known that accurate numerical calculation of fluid flows requires the use of a coordinate system closely aligned with the principal features of the flow. For the nosetip problem this requirement would be satisfied by a coordinate system which closely follows the body shape and, hence, the streamlines of the flow. Because of the asymmetric nosetip geometries being considered, standard coordinate systems (e.g., spherical, axisymmetric reference surface-normal) are incapable of being aligned with the nosetip surface at all points. Thus, a coordinate transformation is sought that will align the coordinate system with an arbitrary nosetip geometry. By requiring the transformation to be in analytic form, the need of solving partial differential equations to define the transformation can be avoided. Finally, the transformation should be in a form that readily lends itself to automated definition, minimizing the inputs required of a user of the code.

## COORDINATE TRANSFORMATION

$(x,y,\phi)$ CYLINDRICAL COORDINATES IN PHYSICAL SPACE

$(\xi,\eta,\theta)$ COORDINATES IN TRANSFORMED SPACE

TRANSFORMATION OF CIRCUMFERENTIAL COORDINATE NOT REQUIRED (NOSETIPS INITIALLY AXISYMMETRIC); ASSUME TRANSFORMATION TAKES THE FORM

$$\xi = \xi\,(x,y,\phi)$$

$$\eta = \eta\,(x,y,\phi)$$

$$\theta = \phi$$

IN A MERIDIONAL PLANE ($\phi$ = CONSTANT), THE TRANSFORMATION REDUCES TO

$$\xi = \xi\,(x,y)$$

$$\eta = \eta\,(x,y)$$

## COORDINATE TRANSFORMATION

The nosetip geometry is defined in an $(x,y,\phi)$ cylindrical coordinate system, and a mapping to a $(\xi,\eta,\theta)$ transformed coordinate system is sought. Since current reentry vehicle nosetips are initially axisymmetric (prior to ablative shape change), it is assumed that nosetip cross-sections retain some "axisymmetric" character during reentry. Thus, no transformation of the circumferential coordinate is required, and $\theta = \phi$ is assigned. (This transformation can readily be generalized to $\theta = f(\phi)$ if required for other applications of this approach.) Within a $\phi$ = constant meridional plane, the transformation reduces to the two-dimensional form $\xi = \xi(x,y)$, $\eta = \eta(x,y)$. Conformal transformations from the $z = x+iy$ to the $\zeta = \xi+i\eta$ plane are desirable, ensuring that an orthogonal $(\xi,\eta)$ grid maps back onto an orthogonal grid in the $(x,y)$ plane.

## DEFINITION OF TRANSFORMATION

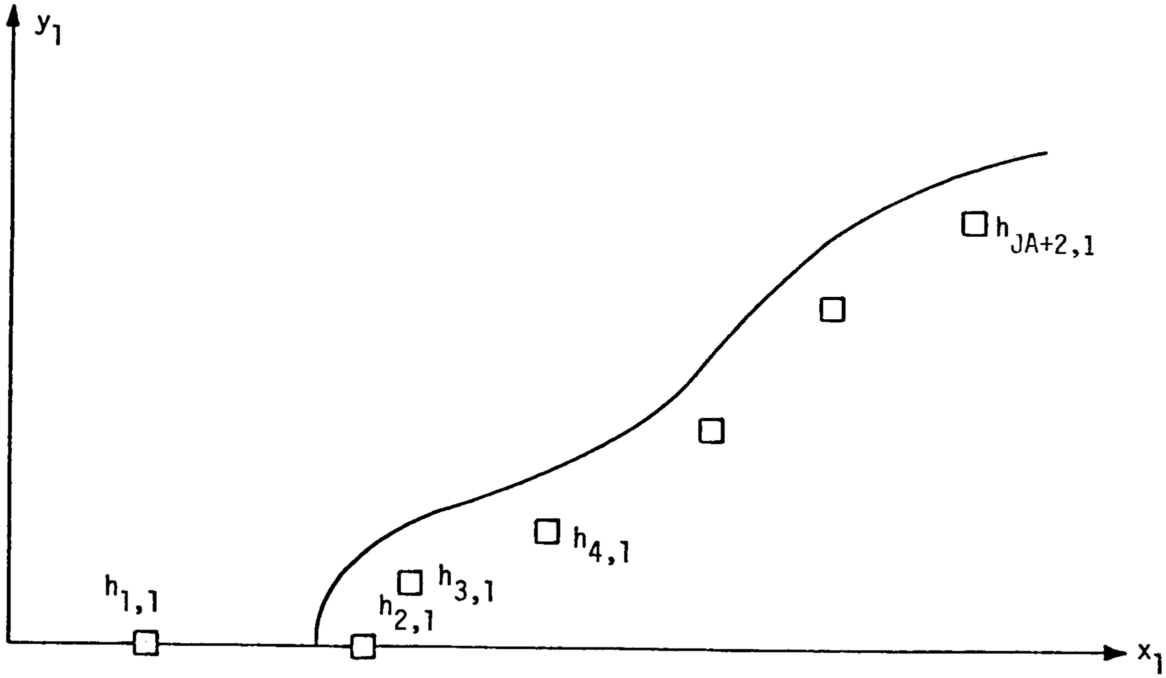
INDEPENDENTLY IN EACH MERIDIONAL PLANE, DEFINE A SEQUENCE OF CONFORMAL TRANSFORMATIONS

$$z_{j+1} - 1 = [z_j - h_{j+1,j}]^{\delta_j} \quad j = 1,2,\ldots,JA$$

$z_j = x_j + i_{yj}$ (j = 1 IS PHYSICAL SPACE)

$h_{i,j}$ = $i^{\underline{th}}$ "HINGE POINT" IN $j^{\underline{th}}$ SPACE

HINGE POINTS ARE SELECTED TO APPROXIMATE BODY GEOMETRY



HINGE POINT DEFINITION

## DEFINITION OF TRANSFORMATION

The approach used to define the coordinate transformations is a modification of the "hinge point" approach of Moretti*. The mapping is defined as a sequence of conformal transformations of the form

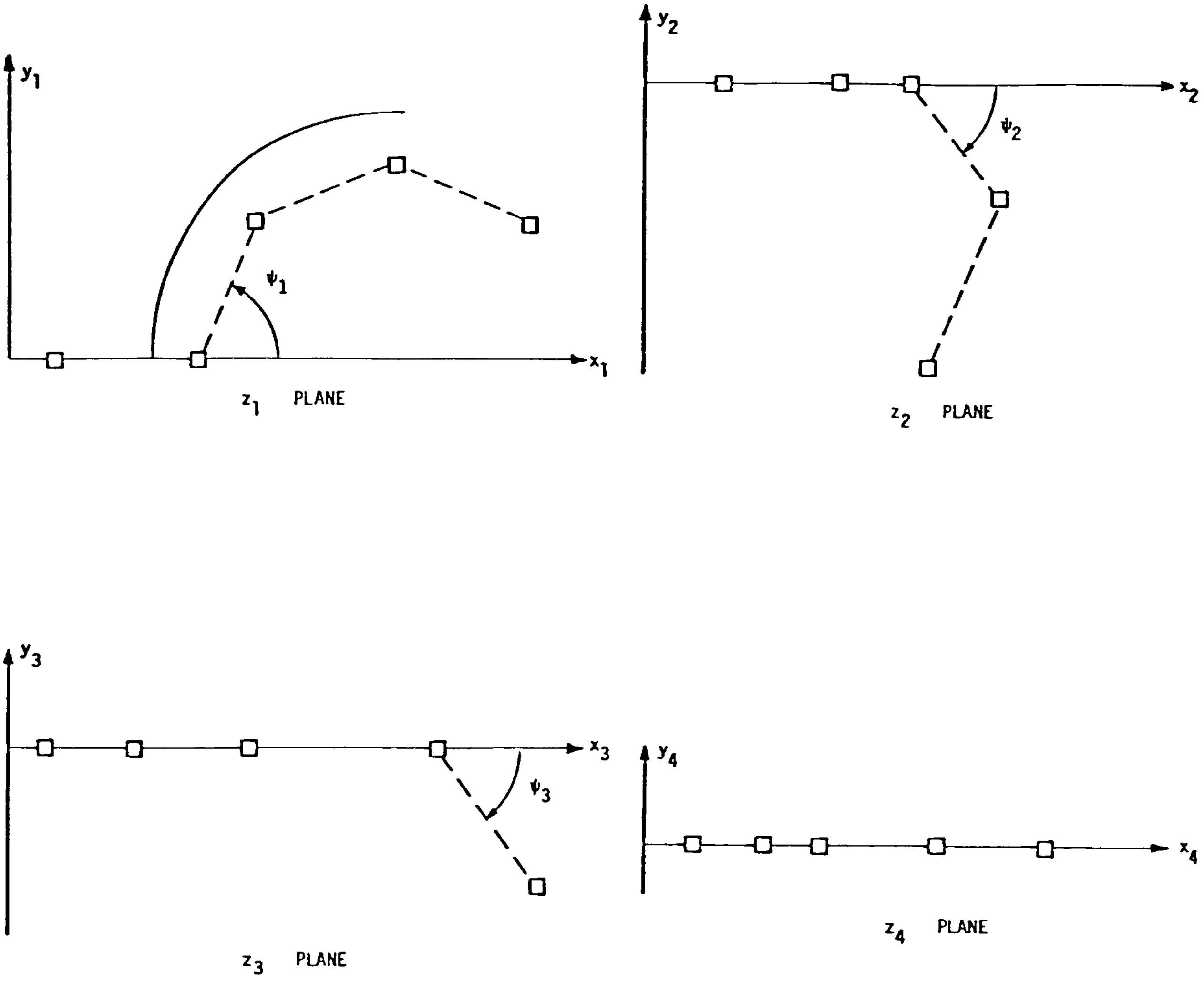
$$z_{j+1} - 1 = [z_j - h_{j+1,j}]^{\delta_j}$$

where $z_j = x_j + iy_j$ ($j = 1$ is physical space) and $h_{i,j}$ is the ith hinge point in the $z_j$ plane. The hinge points in the physical ($z_1$) plane are selected to approximately model the body geometry. By mapping the hinge points sequentially onto the horizontal axis, the image of the body surface will then be a nearly horizontal contour.

-------

*Moretti, G., "Conformal Mappings for Computations of Steady, Three-Dimensional, Supersonic Flows," Numerical/Laboratory Computer Methods in Fluid Mechanics, ASME, 1976.

## SEQUENCE OF TRANSFORMATIONS



$z_1$ PLANE

$z_2$ PLANE

$z_3$ PLANE

$z_4$ PLANE

EXPONENTS OF TRANSFORMATIONS: $\delta_j = \frac{\pi}{\pi-\psi_j}$

# SEQUENCE OF TRANSFORMATIONS

In the j<u>th</u> mapping of the sequence, the transformation is centered around the hinge point $h_{j+1,j}$. The mappings have the property of keeping the hinge points, $h_{i,j}$ $(i \leq j+1)$ on the horizontal axis, while mapping the hinge point $h_{j+2,j}$ onto the horizontal axis. Thus, after JA transformations, all JA+2 hinge points in the JA+1 space will lie on the horizontal axis. (Each mapping in this sequence may be considered a "point-wise Schwarz-Christoffel" transformation.) This figure illustrates the sequence of transformations for JA = 3.

TRANSFORMATIONS - CONTINUED

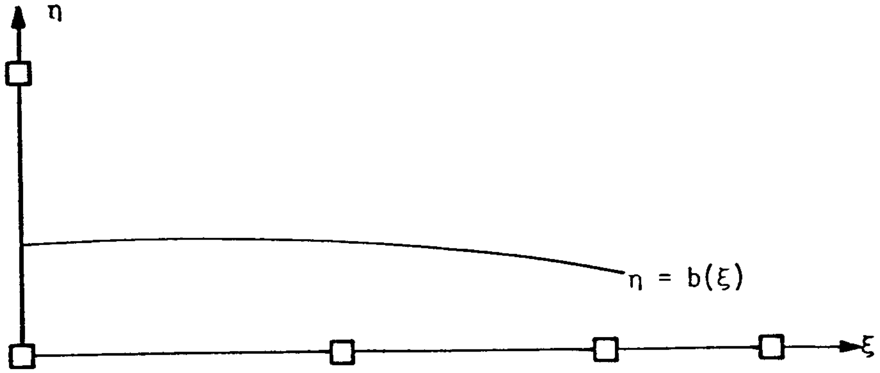
MAP CENTERLINE ONTO VERTICAL AXIS WITH

$$z_{JA+2} = (z_{JA+1} - h_{2,JA+1})^{1/2}$$

ALLOW FOR SIMPLE STRETCHING (REQUIRED FOR CENTERLINE TREATMENT) WITH

$$\zeta = \xi + i\eta = az_{JA+2}$$

RESULTING BODY CONTOUR:



## TRANSFORMATIONS - CONTINUED

In order to establish a grid suitable for flow field calculations when the image of the body contour is a nearly horizontal surface, it is desirable to have the image of the centerline external to the body lie along the vertical axis. This is achieved using an additional conformal transformation, centered around the second hinge point, of the form

$$z_{JA+2} = (z_{JA+1} - h_{2,JA+1})^{1/2} .$$

The last transformation is a simple stretching (which is also conformal):

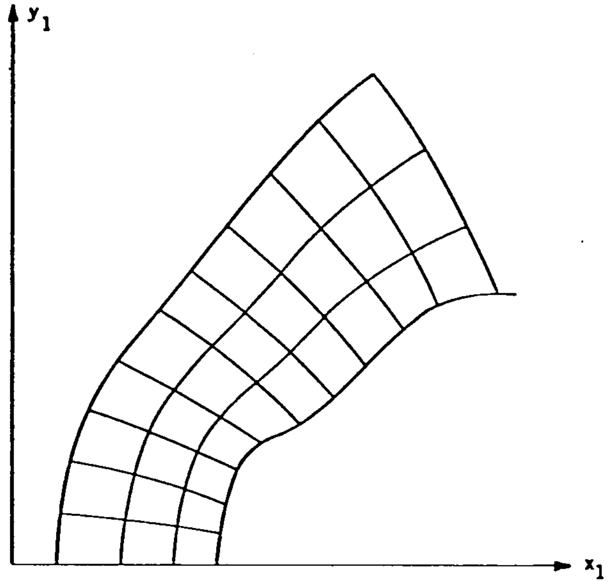
$$\zeta = \xi + i\eta = a z_{JA+2} \quad .$$

(This stretching is used in the calculation procedure along the centerline.) This figure illustrates the body contour resulting in the $\zeta$-plane for the case of a sphere with JA = 3, where the body surface is defined as $\eta = b(\xi)$.

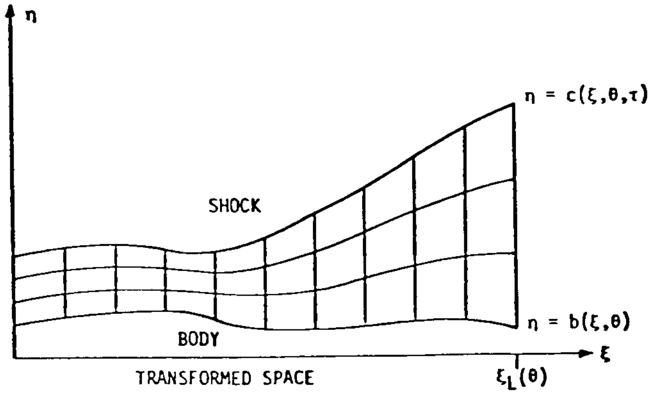
## COMPUTATIONAL TRANSFORMATION

DESIRE GRID POINTS EQUALLY SPACED IN $\xi$ ALONG BODY, IN $\eta$ BETWEEN BODY AND SHOCK, AND IN $\theta$ CIRCUMFERENTIALLY

$$X = \frac{\theta}{2\pi} \qquad Y = \frac{\xi}{\xi_L(\theta)} \qquad Z = \frac{\eta - b(\xi,\theta)}{c(\xi,\theta,\tau) - b(\xi,\theta)}$$



PHYSICAL SPACE



TRANSFORMED SPACE

## COMPUTATIONAL TRANSFORMATION

For the flow field calculation it is desirable to have equally spaced grid points. Thus, a transformation to a computational coordinate system (X,Y,Z) is used, in which grid points are equally spaced circumferentially in $\theta$, longitudinally in $\xi$ within each meridional plane, and in $\eta$ between the body and the shock. It is important to note that the (X,Y,Z) system is not orthogonal, and that the computational transformation varies with time as the bow shock position varies during the time-dependent calculation. These sketches illustrate the computational grids resulting in a meridional plane in both physical ($z = x+iy$) and transformed ($\zeta = \xi+i\eta$) space for a typical ablated nosetip contour (with the shock layer thickness exaggerated for clarity).

## PARAMETERS OF THE TRANSFORMATION

REQUIRED IN WRITING GOVERNING EQUATIONS IN TRANSFORMED COORDINATES

$$g = \frac{\partial \zeta}{\partial z} = G e^{i\omega} = \xi_x + i\eta_x = -i\xi_y + \eta_y$$

$$\Phi = \frac{\partial (\log g)}{\partial \zeta}$$

CAN BE EVALUATED ANALYTICALLY

## CIRCUMFERENTIAL PARAMETERS OF THE TRANSFORMATION

$\zeta_\phi, g_\phi$ CAN BE EVALUATED ANALYTICALLY IF EACH MERIDIONAL PLANE HAS THE SAME NUMBER OF HINGE POINTS, ASSUMING INTERPOLATING FUNCTIONS FOR $h_{i,j}(\phi)$

ALTERNATIVELY, EVALUATE FROM TAYLOR SERIES EXPANSIONS:

$$\zeta_\phi = \xi_\phi + i\eta_\phi = \frac{\zeta_2 - \zeta_1 - g(z_2 - z_1)}{\phi_2 - \phi_1}$$

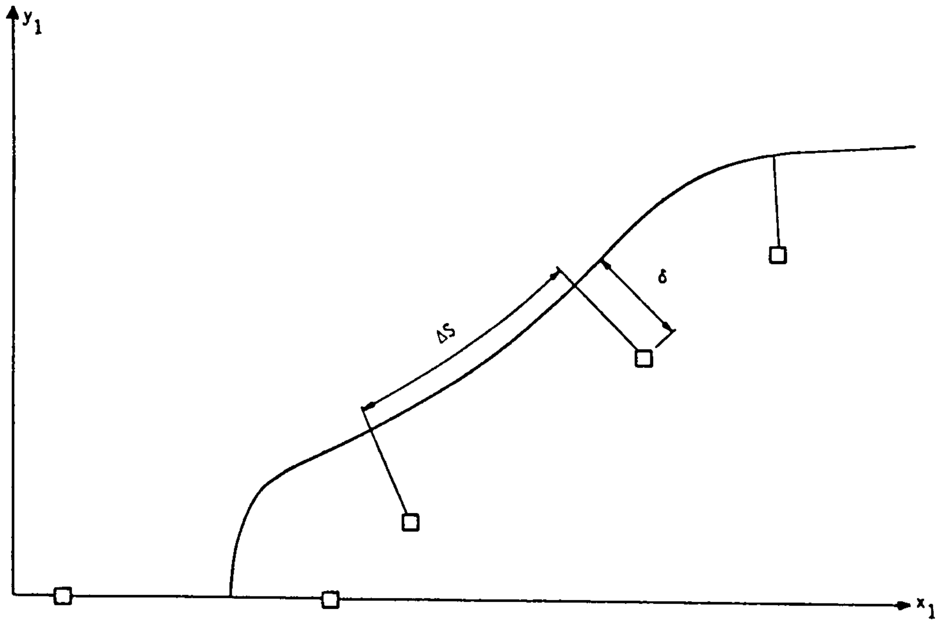
$$g_\phi = \frac{g_2 - g_1 - g^2\ \Phi(z_2 - 2_1)}{\phi_2 - \phi_1}$$

$(\ \ )_1 \rightarrow (X-\Delta X, Y, Z)$, $\quad (\ \ )_2 \rightarrow (X+\Delta X, Y, Z)$ IN COMPUTATIONAL MESH

## PARAMETERS OF THE TRANSFORMATION

In transforming the governing equations from physical to the (X,Y,Z) computational coordinates, certain derivatives of the transformation are required. Because the transformation has been defined in analytic form, these derivatives can readily be evaluated analytically and are functions only of the hinge point locations. Within a meridional plane ($\phi$ = constant), the required derivatives are $g = \partial\zeta/\partial z$ and $\Phi = \partial(\log g)/\partial\zeta$. Circumferentially, the independent transformations in each meridional plane can be coupled to produce a three-dimensional transformation by assuming that hinge point locations can be expressed as $h_{i,j}(\phi)$. The required circumferential parameters of the transformation, $\zeta_\phi$ and $g_\phi$, can be evaluated analytically if each meridional plane has the same number of hinge points and assuming the form of interpolating functions for $h_{i,j}(\phi)$. Alternatively, it has been found to be sufficient to evaluate $\zeta_\phi$ and $g_\phi$ from Taylor series expansions using data at computational (X,Y,Z) mesh points, with the forms of the resulting expressions shown in the figure.

# AUTOMATIC GENERATION OF HINGE POINTS



HINGE POINTS LOCATED DISTANCE $\delta$ ALONG INWARD BODY NORMALS, FROM BODY POINTS EQUALLY SPACED IN WETTED LENGTH

ONLY INPUT REQUIRED OF USER IS NUMBER OF HINGE POINTS TO BE USED IN EACH MERIDIONAL PLANE

## AUTOMATIC GENERATION OF HINGE POINTS

To simplify the application of this coordinate transformation to the asymmetric nosetip flow field problem, the selection of hinge points that define the transformations has been automated. Within each meridional plane to be computed, body normals are constructed at points equally spaced in wetted length along the body profile. The hinge points are then selected to lie a distance $\delta$ inside the body along these normals. By relating $\delta$ to any convenient scale factor for a nosetip geometry, the only input required of the user of the code is the number of hinge points to be used. The locations of the first two hinge points (i.e., those that lie on the x axis) are the same in each meridional plane, in order to simplify the treatment of the centerline. Typically, no more than nine hinge points per meridional plane (JA = 7) are necessary for the nosetip flow field problem.

## TREATMENT OF CENTERLINE

AT THE CENTERLINE ($y = 0$), SCALE FACTORS ($g = \partial\zeta/\partial z$) VARY WITH $\phi$

STRETCHING TRANSFORMATION USED TO MINIMIZE DISCREPANCIES, WITH

$$a(\phi_k) = \frac{h_{1,JA+2}(\phi = 0)}{h_{1,JA+2}(\phi = \phi_k)}$$

CARTESIAN COORDINATES ($x_1, x_2, x_3$) USED IN CENTERLINE ANALYSIS

$$\frac{\partial}{\partial x_1} = \frac{\partial}{\partial x}$$

$$\frac{\partial}{\partial x_2} = \cos\phi \frac{\partial}{\partial y} - \frac{\sin\phi}{y} \frac{\partial}{\partial \phi}$$

$$\frac{\partial}{\partial x_3} = \sin\phi \frac{\partial}{\partial y} + \frac{\cos\phi}{y} \frac{\partial}{\partial \phi}$$

WITH $\lim_{y \to 0} \frac{1}{y} \frac{\partial}{\partial \phi} = \frac{\partial^2}{\partial y \partial \phi}$ FINITE,

$$\frac{\partial}{\partial x_2} = \cos\phi \frac{\partial}{\partial y}, \quad \phi = 0, \pi$$

$$\frac{\partial}{\partial x_3} = \sin\phi \frac{\partial}{\partial y}, \quad \phi = \frac{\pi}{2}, \frac{3\pi}{2}$$

## TREATMENT OF CENTERLINE

The greatest complication encountered in the use of this 3-D coordinate transformation is the extra care that must be taken in treating the grid points on the centerline. Since the transformations in each meridional plane are independent, the scale factors $g = \partial\zeta/\partial z$ along the centerline will not be the same in each meridional plane. Thus, one computational grid point at the centerline will represent different physical points for each value of $\phi$ . To minimize these discrepancies, the stretching transformation $\zeta = az_{JA+2}$ is used to ensure that the images of the first hinge point are coincident in all meridional planes. The remaining discrepancies are small enough that simple linear interpolations can be used to account for differences in the scale factors.

In addition to the mapping complications along the centerline, the governing equations in cylindrical coordinates are singular along $y = 0$. This difficulty has been avoided by using a Cartesian $(x_1,x_2,x_3)$ coordinate system for the centerline analysis. The required Cartesian derivatives can be expressed in terms of the radial derivative $\partial/\partial y$ in cylindrical coordinates for certain values of $\phi$, as shown in this figure. The only restriction resulting from this analysis is that computational planes must be located at $\phi = 0$, $\pi/2$, $\pi$, and $3\pi/2$.

## RESULTING FLOW FIELD CODE

CM3DT (CONFORMAL MAPPING 3-D TRANSONIC)

NOSETIP FLOW FIELD CODE

- IDEAL OR EQUILIBRIUM REAL GAS THERMODYNAMICS
- PITCH AND YAW CAPABILITY
- $\lambda$-DIFFERENCING SCHEME USED TO TREAT WEAK EMBEDDED SHOCKS ON INDENTED NOSETIPS
- COUPLED TO AFTERBODY CODES FOR TOTAL INVISCID FLOW FIELD CAPABILITY
  - BMO/3IS
  - NSWC/D3CSS
  - STEIN
- $81,000_{10}$ CORE STORAGE REQUIRED

## RESULTING FLOW FIELD CODE

The 3-D, time-dependent, inviscid nosetip flow field code that was developed using the 3-D coordinate transformation described here is called CM3DT (Conformal Mapping 3-D Transonic). This code can treat ideal or equilibrium real gas thermodynamics, has both pitch and yaw capability, and is able to treat weak embedded shocks on indented nosetips using the $\lambda$-differencing scheme*. To provide total body inviscid flow field capability, the CM3DT code has been coupled to the BMO/3IS**, NSWC/D3CSS[+], and STEIN[++] afterbody codes. Complete details on the CM3DT analysis and results obtained with this code may be found in the following references:

Hall, D. W., "Inviscid Aerodynamic Predictions for Ballistic Reentry Vehicles with Ablated Nosetips," Ph.D. Dissertation University of Pennsylvania, 1979; also, BMO TR to be published.

Hall, D. W., "Calculation of Inviscid Supersonic Flow over Ablated Nosetips," AIAA Paper 79-0342, January 1979.

---------------

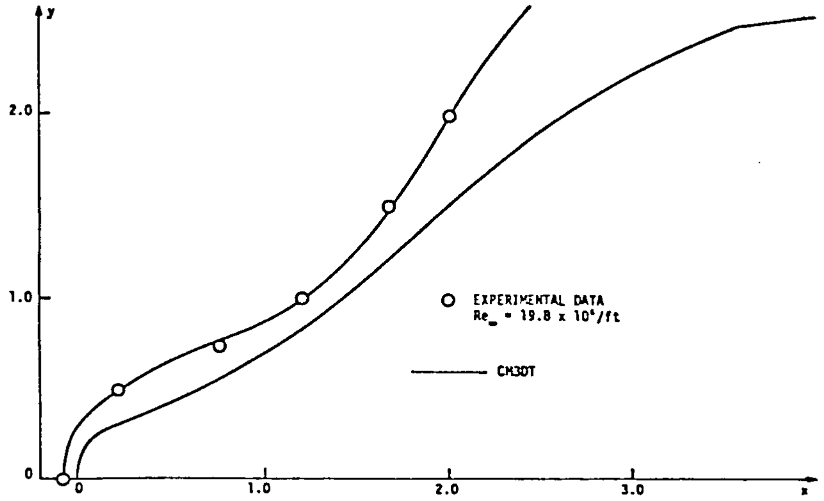
*Moretti, G., "An Old Integration Scheme for Compressible Flow Revisited, Refurbished, and Put to Work," Polytechnic Institute of New York, POLY-M/AE Report 78-22, September 1978.

**Kyriss, C. L. and Harris, T. B., "A Three-Dimensional Flow Field Computer Program for Maneuvering and Ballistic Reentry Vehicles," 10th USNavy Symposium on Aeroballistics, July 1975; also, Daywitt, J., Brant, D., and Bosworth, F., "Computational Technique for Three-Dimensional Inviscid Flow Fields about Reentry Vehicles, Volume I: Numerical Analysis," SAMSO TR-79-5, April 1978.

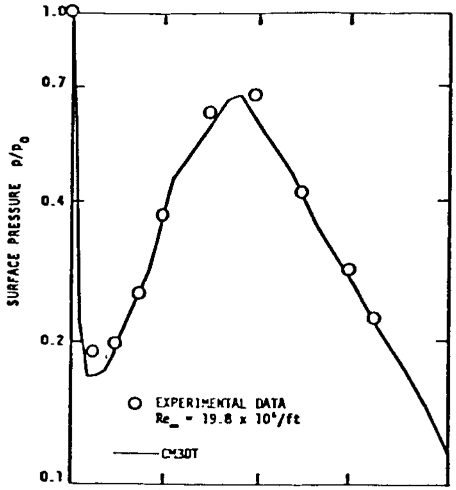
[+] Solomon, J. M., Ciment, M., Ferguson, R. E., Bell, J. B., and Wardlaw, A. B., Jr., "A Program for Computing Steady Inviscid Three-Dimensional Supersonic Flow on Reentry Vehicles, Volume I: Analysis and Programming," Naval Surface Weapons Center, NSWC/WOL/TR 77-28, February 1977.

[++] Marconi, F., Salas, M., and Yaeger, L., "Development of a Computer Code for Calculating the Steady Super/Hypersonic Inviscid Flow around Real Configurations, Volume I. Computational Technique," NASA CR-2675, April 1976.

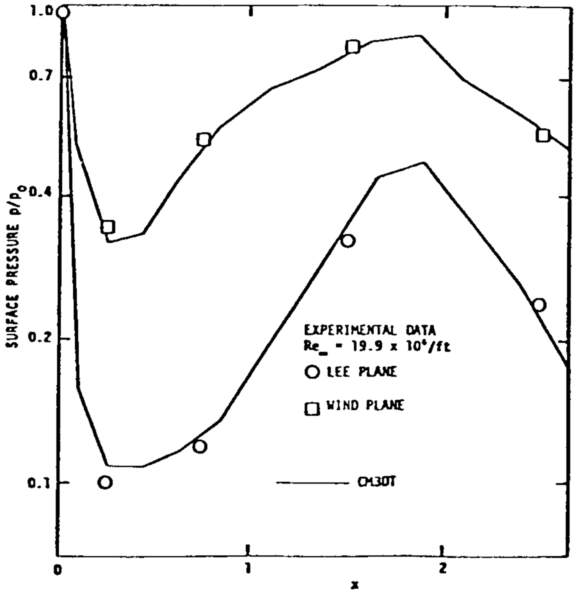
## CM3DT RESULTS



CM3DT Shock Shape Prediction for the PANT Triconic at $M_\infty = 5$, $\alpha = 0°$



Surface Pressure Predictions for the PANT Triconic at $M_\infty = 5$, $\alpha = 0°$



Surface Pressure Predictions for the PANT Triconic at $M_\infty = 5$, $\alpha = 10°$

# CM3DT RESULTS

This figure presents some typical results obtained with the CM3DT inviscid nosetip flow field code. Shown are comparisons of predictions to data obtained for the PANT Triconic shape* at $M_\infty = 5$. It is significant that attempts to compute the flow over this slender shape using a time-dependent code formulated in a spherical coordinate system were unsuccessful. CM3DT, with its body-oriented coordinate system, was able to obtain converged solutions for this shape, with the predictions agreeing well with the data, as seen in this figure.

-------

*Abbett, M. J. and Davis, J. E., "Interim Report, Passive Nosetip Technology (PANT) Program, Volume IV. Heat Transfer and Pressure Distribution on ablated Shapes, Part II. Data Correlation and Analysis," Space and Missile Systems Organization, TR-74-86, January 1974.

## CM3DT - RUN TIMES

ON A CDC CYBER 176, CM3DT REQUIRES 0.00045 CP SECS/POINT/STEP FOR IDEAL GAS CALCULATIONS WITH $\lambda$-DIFFERENCING

- 20% PENALTY INCURRED FOR COORDINATE TRANSFORMATION (PARAMETERS ON MOVING GRID UPDATED EVERY 10 TIME STEPS)

- 50% PENALTY INCURRED FOR $\lambda$-DIFFERENCING (RELATIVE TO MAC CORMACK DIFFERENCING)

## CM3DT - RUN TIMES

On a CDC Cyber 176 computer, the CM3DT inviscid nosetip code with λ-differencing requires approximately 0.00045 CP seconds per grid point per time step (iteration). Typically, 400-500 time steps are required to obtain a converged solution. It is estimated that the computer time required for a solution has been increased by approximately 20% by using the 3-D coordinate transformation described here, when the parameters of the transformation on the moving grid are updated every ten time steps. When compared to the standard MacCormack differencing scheme, the use of λ-differencing scheme increases the run time requirements approximately 50% for this code.

# Application of the Multigrid Method to Grid Generation

by

Samuel Ohring

Computation, Mathematics, and Logistics Department
David W. Taylor Naval Ship Research and Development Center
Bethesda, Maryland 20084

# ABSTRACT

The multigrid method (MGM) has been used to numerically solve the pair of nonlinear elliptic equations commonly used to generate two-dimensional boundary-fitted coordinate systems. Two different geometries are considered: one involving a coordinate system fitted about a circle and the other selected for an impinging jet flow problem. MGM uses a nest of grids from finest (upon which the solution is sought) to coarsest and is based on the idea of using relaxation sweeps to smooth the error (equivalent to eliminating high frequency Fourier components of the error). Thus most of the computational work is done on coarser subgrids to eliminate longer wave length components of the error. Two different relaxation schemes are tried: one is successive point overrelaxation and the other is a four-color scheme vectorizeable to take advantage of a parallel processor computer for greater computational speed. Results using MGM are compared with those using SOR (doing successive overrelaxations with the corresponding relaxation scheme on the fine grid only). It is found that MGM becomes significantly more effective than SOR as more accuracy is demanded and as more corrective grids, or more grid points, are used. For the accuracy required here, it is found that MGM is two to three times faster than SOR in computing time. With the four-color relaxation scheme as applied to the impinging jet problem the advantage of MGM over SOR is not as great. Perhaps this is due to the effect of a poor initial guess on MGM for this problem.

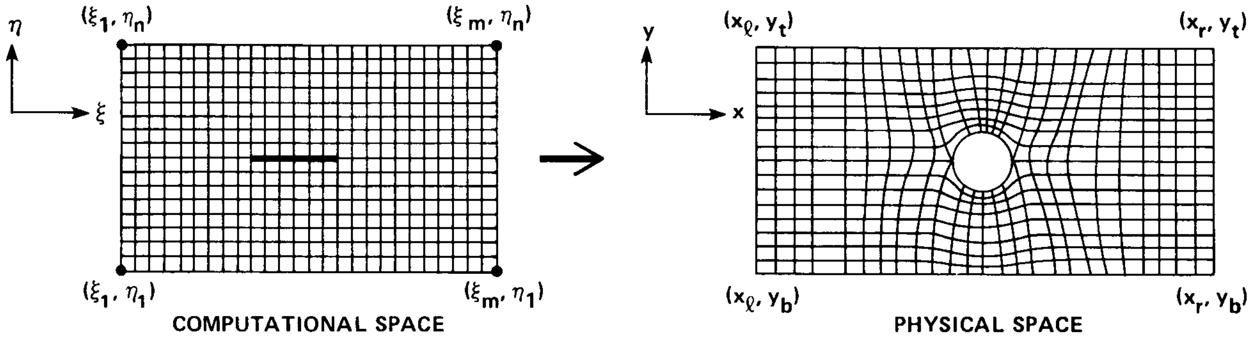

$\eta$
$(\xi_1, \eta_n)$
$(\xi_m, \eta_n)$
$\xi$
$(\xi_1, \eta_1)$
$(\xi_m, \eta_1)$
COMPUTATIONAL SPACE
y
$(x_\ell, y_t)$
$(x_r, y_t)$
x
$(x_\ell, y_b)$
$(x_r, y_b)$
PHYSICAL SPACE

Figure 1

The multigrid method (MGM) [1] can numerically solve linear or nonlinear elliptic partial differential equations more rapidly than conventional means of solution such as successive overrelaxation (SOR). MGM can be applied to the numerical solution of partial differential equations not amenable to numerical solution by fast direct matrix solvers such as diagonal decomposition. Thus it was deemed desirable to apply MGM to the numerical solution of the system of nonlinear elliptic equations commonly used to generate boundary-fitted coordinate systems, especially when the number of grid points is large. The standard elliptic equations for a typical mapping, shown schematically in Figure 1, are

$$L_1(x,y) = \alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_\xi + Qx_\eta) = 0 \tag{1}$$

$$L_2(x,y) = \alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_\xi + Qy_\eta) = 0 \tag{2}$$

where

$$\begin{aligned} \alpha &= x_\eta^2 + y_\eta^2 & \beta &= x_\xi x_\eta + y_\xi y_\eta \\ \gamma &= x_\xi^2 + y_\xi^2 & J &= x_\xi y_\eta - x_\eta y_\xi \end{aligned} \tag{3}$$

and P and Q are functions of $\xi$ and $\eta$. Dirichlet conditions are specified on all boundaries of the computational space including the interior slit (which maps to the body in the physical space). Each side of the slit has a set of Dirichlet data with a common value for each of the endpoints of the slit.

[1] A. Brandt, Math. of Comp., Vol. 31, No. 138, April 1977, pp. 333-390.

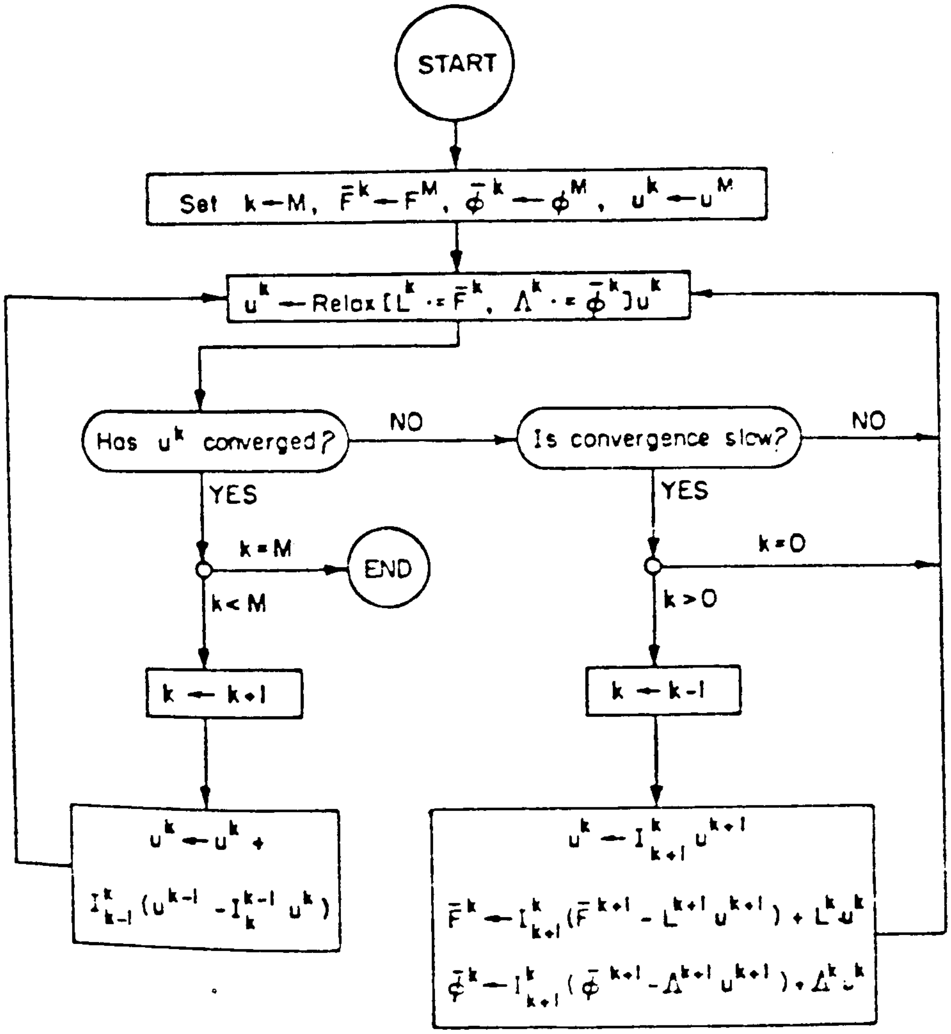

START
Set $k \leftarrow M$, $\bar{F}^k \leftarrow F^M$, $\bar{\phi}^k \leftarrow \phi^M$, $u^k \leftarrow u^M$
$u^k \leftarrow \text{Relax}[L^k \cdot = \bar{F}^k, \Lambda^k \cdot = \bar{\phi}^k]u^k$
Has $u^k$ converged?
NO
Is convergence slow?
NO
YES
$k = M$
END
$k < M$
YES
$k = 0$
$k > 0$
$k \leftarrow k+1$
$k \leftarrow k-1$
$u^k \leftarrow u^k +$
$I^k_{k-1}(u^{k-1} - I^{k-1}_k u^k)$
$u^k \leftarrow I^k_{k+1} u^{k+1}$
$\bar{F}^k \leftarrow I^k_{k+1}(\bar{F}^{k+1} - L^{k+1} u^{k+1}) + L^k u^k$
$\bar{\phi}^k \leftarrow I^k_{k+1}(\bar{\phi}^{k+1} - \Lambda^{k+1} u^{k+1}) + \Lambda^k u^k$

Figure 2

The basic idea of MGM is to do most of the computational work on coarser corrective grids containing far fewer points than the finest grid upon which the solution is sought. The grids form a nest, each coarser grid having twice the mesh spacing in each coordinate direction of the previous finer grid. In Figure 2 which represents the Full Approximation Storage scheme of [1]: u = (x,y), $L = \begin{Bmatrix} L_1 \\ L_2 \end{Bmatrix}$ such that Eqs. (1) and (2) become $Lu = F = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$, $1 \leq k \leq M$ (k representing the kth grid with M the finest), $\phi = (x,y)$ on the boundaries of the computational space (Dirichlet values so that $\Lambda$ is an identity operator) and superscripts refer to discretized quantities on the kth grid. (All operations involving $\phi$ in the flow chart can be ignored, since the Dirichlet conditions are constant on all the grids.) The main idea behind MGM is that relaxation sweeps are a smoothing process which eliminate the highest frequency Fourier components of the error on any grid. First, starting with an initial guess for the solution, several sweeps are carried out on the finest grid to eliminate high frequency components of the error. The smoothed out error is represented by the residual $f^M = L^M U^M - L^M u^M$ and the correction $U^M - u^M$ (where $U^M$ is the exact discrete solution on the finest Mth grid). The residual, consisting mainly of longer wave-length Fourier components, is dealt with by solving its coarser-grid approximation

$$L^{M-1} U^{M-1} - L^{M-1} I_M^{M-1} u^M = I_M^{M-1} f^M \tag{4}$$

for $U^{M-1}$, which is represented by $\overline{F}^k$ for $k = M-1$ in the lower right box of Figure 2. The symbol $I_k^{k-1}$ means interpolation of a quantity from the kth grid to the (k-1)st grid. Eq. (4) is solved in the same way as the original equation on the finest grid. If solution of (4) is obtained after several relaxation sweeps, the coarse grid approximation $U^{M-1} - I_M^{M-1} u^M$ to the smoothed out function $U^M - u^M$ is added to $u^M$. That is $u^M \leftarrow u^M + I_{M-1}^M (u^{M-1} - I_M^{M-1} u^M)$, which is the expression in the lower left box for $k = M$. The new $u^M$ is a better approximation to the solution $U^M$ and is the starting point for more relaxation sweeps for the original set of Eqs. (1) and (2) on the finest grid. If convergence is obtained, the process is complete; if not, the process returns to the coarser grid to sweep the residual equation again. If it doesn't converge after a few sweeps, then the next coarser grid is used to eliminate long wave length errors for the residual equation, etc. Each residual equation has a corresponding residual equation and correction on the next coarser grid. (The residuals were weighted locally as in [1].)

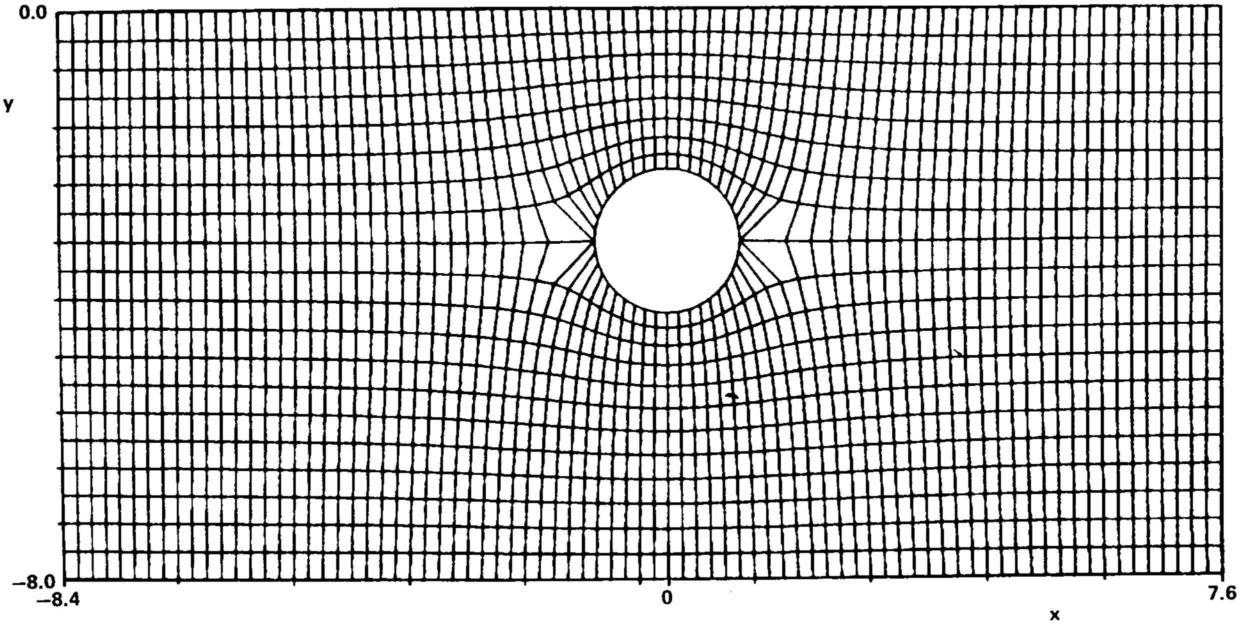


Figure 3

Figure 3 shows computer drawn body-fitted coordinate systems generated to a specific accuracy using MGM and SOR (the two coordinate systems coincide). The relaxation scheme used was successive point overrelaxation. According to the notation used in Figure 1, m and n are 81 and 21, respectively; the slit end points are $(\xi_{33},\eta_{13})$ and $(\xi_{53},\eta_{13})$, respectively; $(x_\ell,y_b) = (-8.4, -8.0)$ and $(x_r,y_t) = (7.6, 0.0)$; $\Delta x = .2$ and $\Delta y = .4$; $\Delta\xi = \Delta\eta = 1$; and the body is a circle of radius one centered at $(x,y) = (0, -3.2)$. P and Q were set to zero in Eqs. (1) and (2). An experimentally determined, essentially optimum overrelaxation factor of 1.7 was used in the successive point overrelaxation sweeps in both MGM and the SOR method. All coarser corrective grids contain grid points on the slit. The initial guess for $x(\xi,\eta)$, $y(\xi,\eta)$ in the computational space is obtained by extending the Dirichlet data at the outer boundaries throughout the space except at the slit, where the body Dirichlet data are used. The convergence criterion for the solution of Eqs. (1) and (2) was that both $L_2$-error norms (one for each equation) be less than an input value $||E||_{L_2}$. (This will be called satisfaction of $||E||_{L_2}$.) For Figure 3, $||E||_{L_2} = .001$. To satisfy this criterion, MGM used 32.5 WU and 16.08 CP seconds compared to 66.0 WU and 22.17 CP seconds for SOR. (A work unit (WU) is the equivalent of one SOR sweep on the finest grid, and CP seconds refer to central processor seconds used on the Texas Instruments Advanced Scientific Computer (TI-ASC).) For $||E||_{L_2} = .01$, MGM used 20 WU compared to 29 WU for SOR; CP time was the same for both methods (due mainly to the additional computational work in computing residuals in MGM). The results show that the effectiveness of MGM increases (compared to SOR) as the error norm decreases. This is consistent with the fact that the remaining longer wave length errors are eliminated more slowly using SOR. The parameters $\delta = .3$, $\zeta = .3$ were used to control the flow of MGM. The parameter $\delta$ determines the convergence test on each grid and the parameter $\zeta$ determines how fast the convergence must be (how fast the high frequency components are eliminated) on each grid. Whenever $\zeta < (||E||^k_{L_2})^{i+1}/(||E||^k_{L_2})^i$ on a kth grid, MGM will then process on the coarser (k-1)st grid with an error norm to be satisfied equal to $\delta(||E||^k_{L_2})^{i+1}$. (Superscripts i,k refer to the ith relaxation sweep and the kth grid, respectively.) These parameters are used as in [1], have a range ($0 < \delta < 1$; $0 < \zeta < 1$), and greatly influence the performance of MGM. The present choice is not necessarily optimum but was the best of a number of choices tried in the unit square.

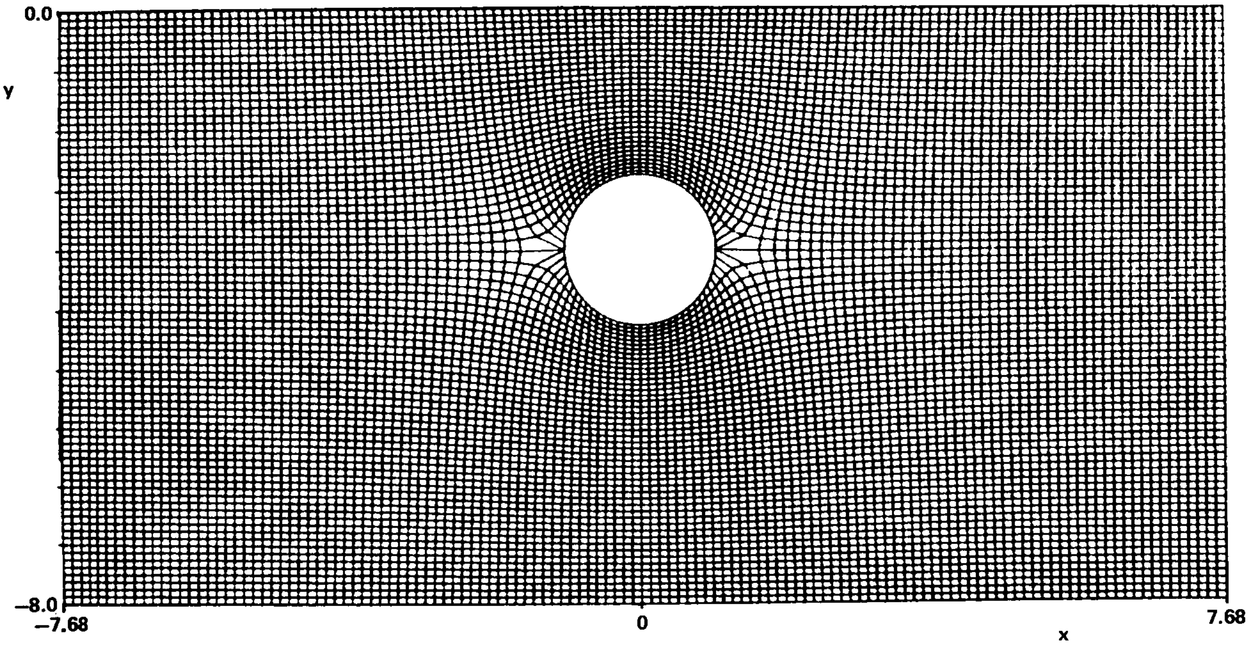


Figure 4

Figure 4 shows a computer drawn body-fitted coordinate system, similar to Figure 3, generated with MGM and satisfying $||E||_{L_2} = .001$. The grid parameters are (see Figure 1): m and n equal to 129 and 81, respectively; slit end points of $(\xi_{49}, \eta_{49})$ and $(\xi_{81}, \eta_{49})$, respectively; $(x_\ell, y_b) = (-7.68, -8.0)$ and $(x_r, y_t) = (7.68, 0.0)$; $\Delta x = .12$ and $\Delta y = .1$; $\Delta\xi = \Delta\eta = 1$; and the circle of radius one was centered again at (0, -3.2). To satisfy $||E||_{L_2} = .001$ MGM used 21.863 WU and 70.67 CP seconds compared to 102.0 WU and 217.83 CP seconds used by SOR. This represents a significant saving of computer time by MGM. To satisfy $||E||_{L_2} = .01$ MGM used 10.863 WU compared to 17.0 WU used by SOR with CP time essentially the same. These results, along with those for Figure 3, show that MGM is more effective, compared to SOR, when more corrective grids are used and more accuracy is required. Figure 4 has five corrective grids and Figure 3 has three corrective grids (including the finest). The parameters $\delta = .03$ and $\zeta = .2$ controlled MGM for Figure 4. Choosing smaller $\delta$ and $\zeta$ makes it more likely that all the coarser corrective grids will be used, which is desirable.

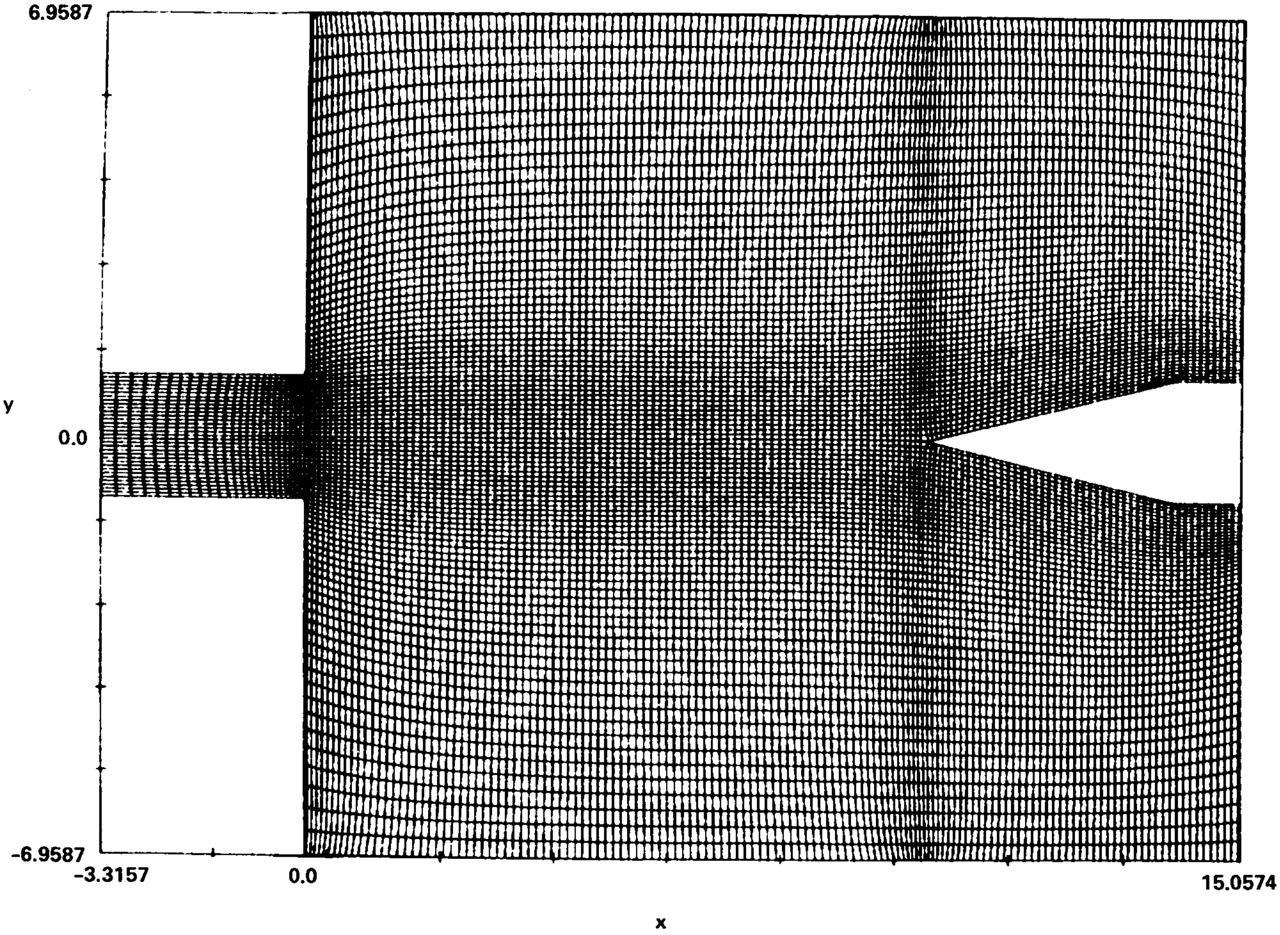


Figure 5

Figure 5 shows a computer drawn body-fitted coordinate system generated using MGM and satisfying $||E||_{L_2} = .001$. SOR was also used to generate this grid and is in excellent agreement with MGM. The geometry is motivated by an impinging jet flow problem that is planned to be run on this grid. The flow from the channel interacts with the solid body on the right. The computational space has the same shape as the physical space except that the body is replaced by a slit. Excluding the channel, the grid consists of 137 points in the horizontal direction by 97 points in the vertical direction. The grid for the channel itself consists of 25 horizontal grid points by 33 vertical grid points. The slit (and body) are 49 grid points long. Corner points on the body and channel have been excluded from the grid. Exponential grid spacing was used along various parts of the horizontal and vertical boundaries of the grid. In an attempt to preserve this boundary spacing in the grid interior non-zero P and Q were used. Although grid lines are still bent near the boundaries, they are not bent as much as when $P = Q = 0$ was tried. To compute this grid (which had 4 corrective grids, including the finest) MGM was "vectorized" on the TI-ASC since it is a parallel processor machine. To accomplish vectorization, which cut computing time by a factor of six, a four-color relaxation scheme was used (i.e., even points of even rows were relaxed simultaneously; odd points of even rows; etc.). With this scheme MGM used 82.781 WU and 45.57 CP seconds to satisfy $||E||_{L_2} = .001$ when using an overrelaxation factor (RF) of 1.8 on the finest grid and relaxation factors of 1.6, 1.4, and 1.2 for the progressively coarser grids. (Varying RF in this way improved MGM's performance.) SOR (with the four-color scheme) used 170.0 WU and 76.68 CP seconds using a relaxation factor of 1.8, which is about optimum for this SOR. MGM used 60.641 WU and 36.67 CP seconds to satisfy $||E||_{L_2} = .001$ when RF's of 1.6, 1.4, 1.2, 1.0 were used on progressively coarser grids (with 1.6 used for the finest grid). With these RF's MGM used 26.016 WU to satisfy $||E||_{L_2} = .01$ compared to 82.0 WU used by SOR with RF = 1.8. The parameters $\delta = .05$, $\eta = .95$ were used for MGM which was divergent for $\eta < .9$. MGM should perform better with a better initial guess than used here. (The horizontal straight lines in the initial guess were discontinuous at the right-most boundary.)

FINITE DIFFERENCE MESH MODIFICATION TECHNIQUE WITH APPLICATION TO SUPERSONIC WEDGE FLOW

by

Fred W. Martin

PAPER NOT AVAILABLE FOR PRINTING

## POSTER SESSION II

1. "Curved Finite Element Mesh Verification"
   R. Wait & N. G. Brown

2. "Grid Generation for General Three-Dimensional Configurations"
   K. D. Lee, M. Huang, Neng Jong Yu, P. E. Rubbert

3. "Boundary-Fitted Coordinate System for Arbitrary Computational Regions"
   Edward Kowalski

4. "A Variable Coefficient for Boundary Fitted Coordinate Systems"
   Tom Bowman

5. "Grid & Metric Generation on the Assembly of Local Bi-Quadratic Coordinate Transformations"
   A. J. Baker, M. O. Soliman, P. D. Manhardt

6. "Generations of Orthogonal Surface Coordinates"
   Fred Blottner, J. B. Moreno

7. "Construction of Composite Three-Dimensional Grids from Subregion Grids Generated by Elliptic Systems"
   P. D. Thomas

8. "Some Aspects of Adapting Computational Meshes to Complex Flow Domains and Structures with Applications to Blown Shock Layer and Base Flow"
   Charles K. Lombard, M. P. Lombard, G. P. Menees, J. Y. Yang

9. "Grid Generation Using Course Smooth Finite Elements"
   Lawrence J. Dickson

10. "Generation of Boundary and Boundary-Layer Fitting Grids"
    C. M. Ablow, S. Schechter

11. "Curvilinear Grids for Sinuous River Channels"
    Dr. Frank B. Tatom, P.E., William Waldrop, S. Ray Smith

# Curved finite Element Mesh Verification

R. Wait and N.G. Brown

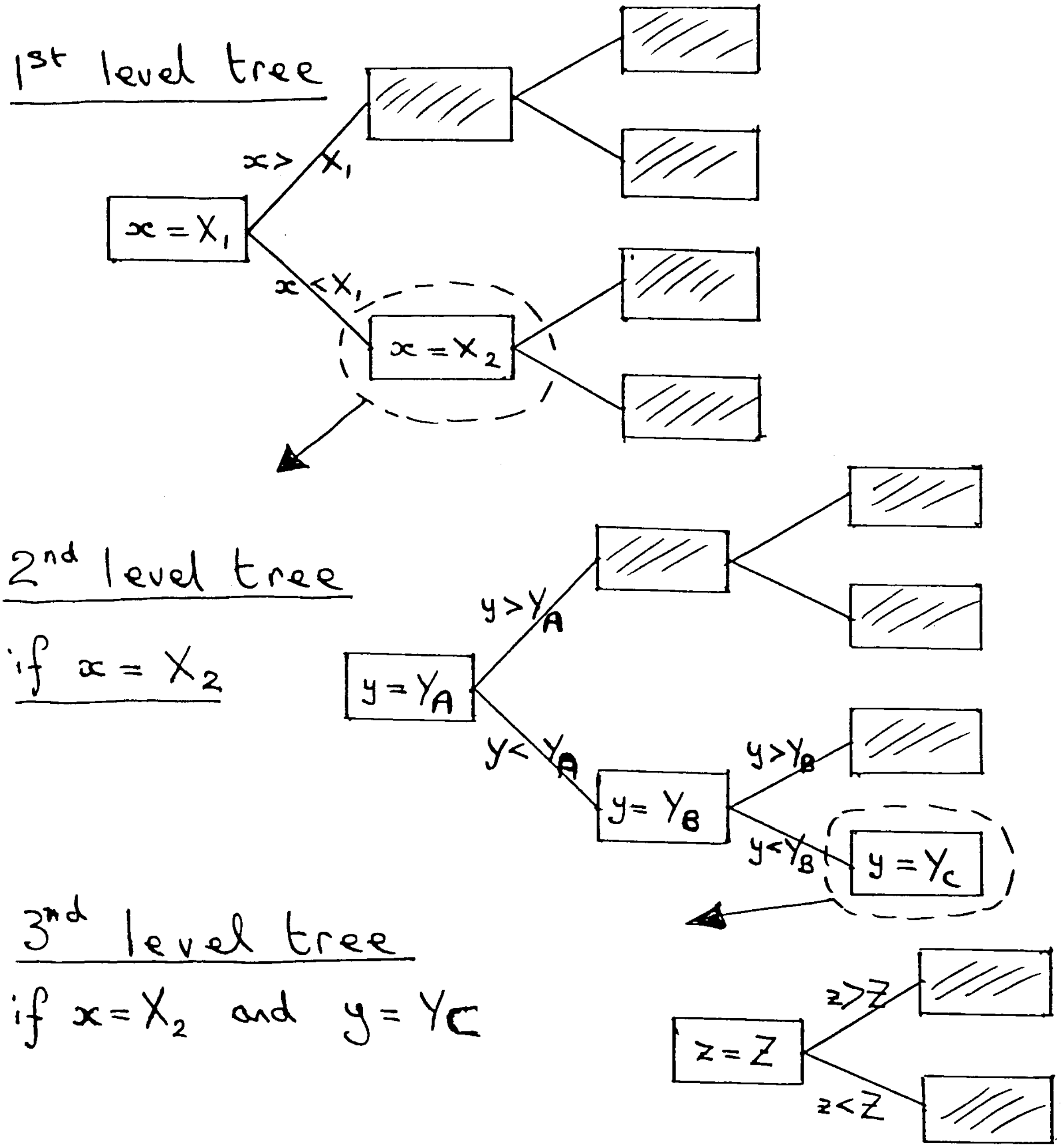
Department of Computational and Statistical Science,
University of Liverpool,
Liverpool L69 3BX,
England.

## Abstract

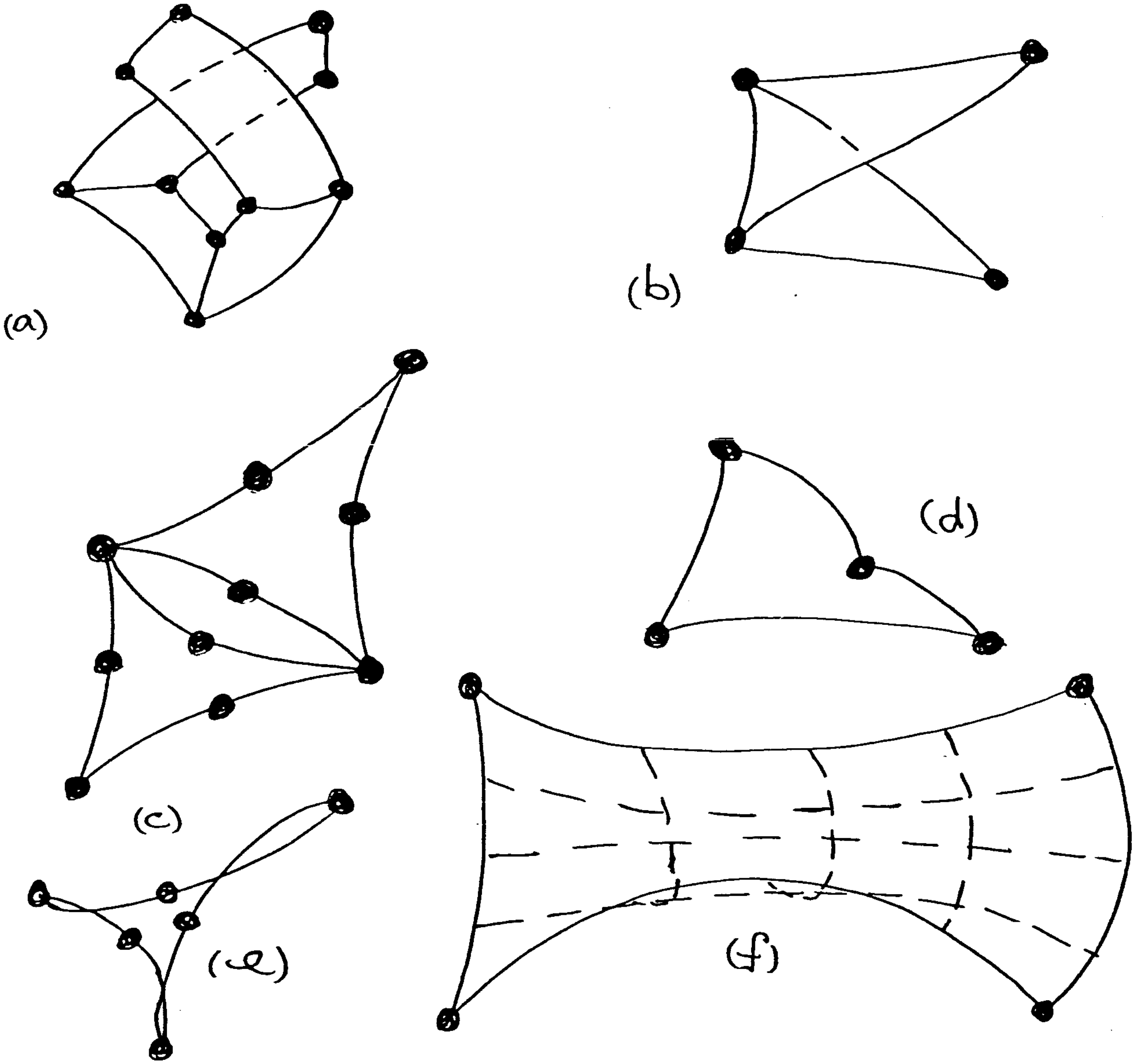
In a computer program a finite element mesh in invariably represented as a set of lists of numbers. It is the aim of this paper to provide techniques whereby it is possible to verify that given sets of data correctly define a well posed finite element grid. Thus we wish to check that the data corresponding to an individual element represents a single region with a simply connected interior. In addition it is necessary to check that when the elements are pieced together there are no overlaps and no gaps left in the representation of the domain of the problem. In addition it is useful to check that nodes are only duplicated or pathologically close together when this is necessary as for example when discretizing one region with a crack.

We consider isoparametic and blended finite elements and provide simple algebraic and/or geometric conditions that have to be satisfied if the transformation on to a reference element is one-one. It can be shown that the well known '$\frac{1}{4}$-point' condition is a special case of one of the necessary conditions but is only sufficient in exceptional cases. Equivalent necessary conditions for three-dimensional tetrahedron and brick elements are developed. Cubic and higher order elements can be included in more general conditions derived from blending function approximations.

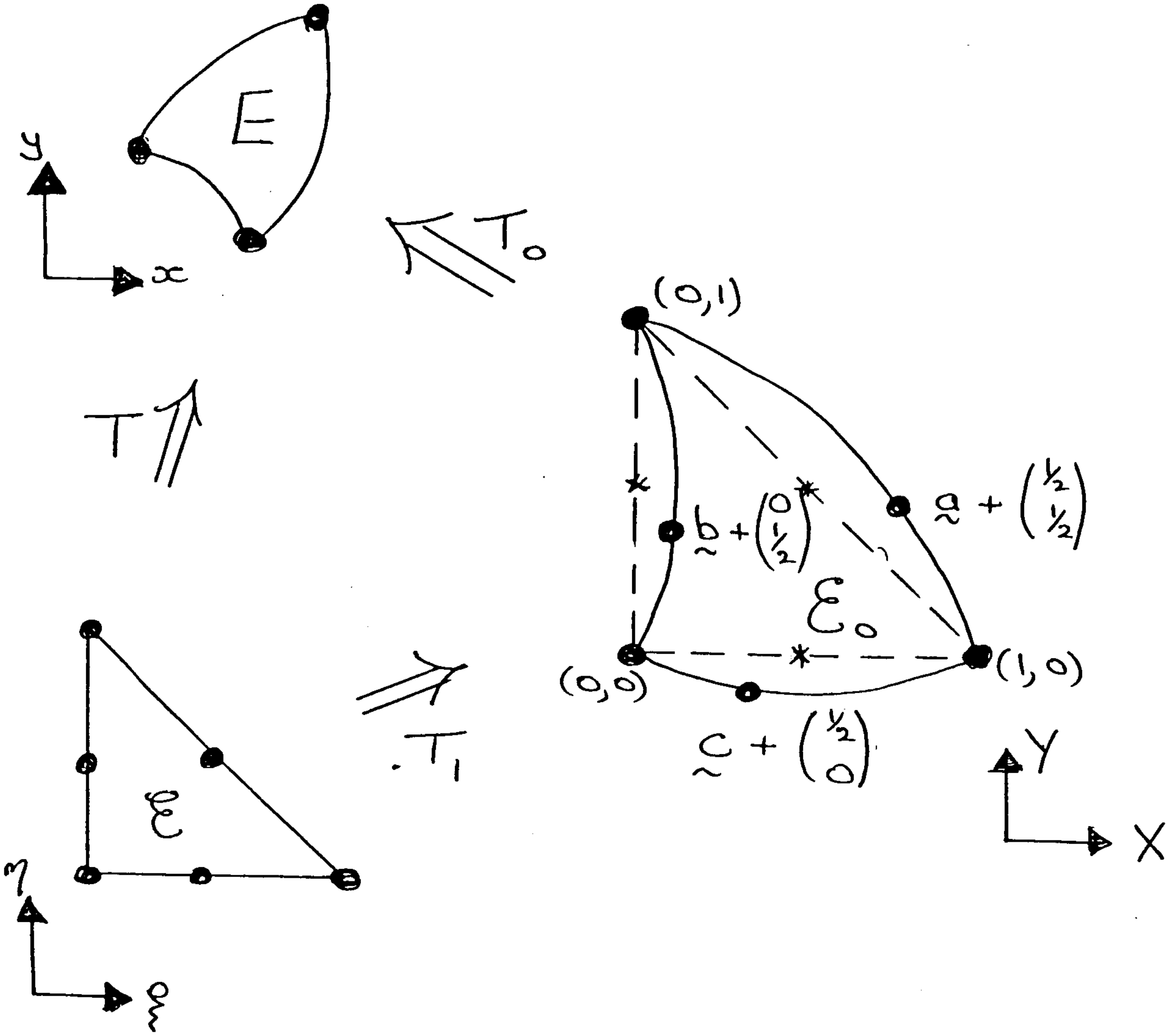
Summary of a poster presentation at "Grid Generation Workshop", NASA-Langley Research Centre, 6-7th October, 1980.



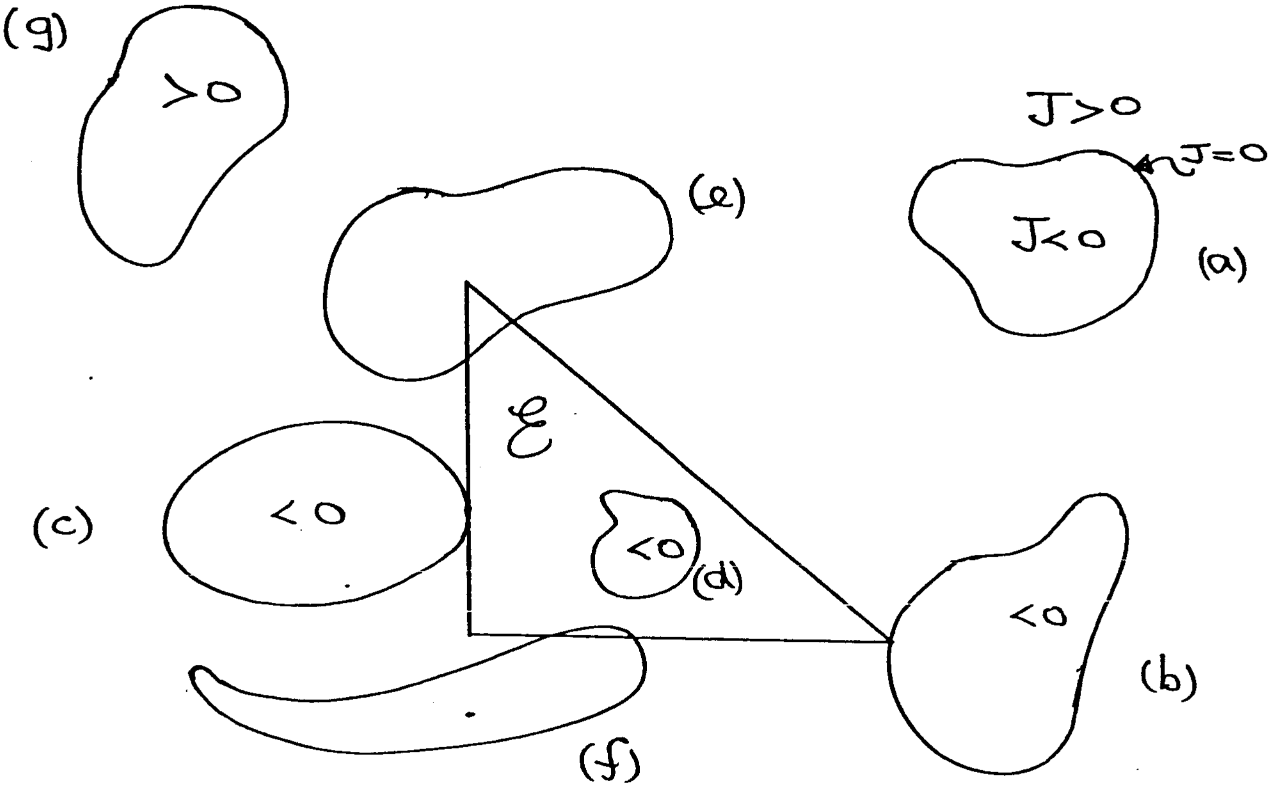
If N is the total number of co-ordinates then any search through the co-ordinates involves O(N log N) operations if the search is arranged in terms of nested binary trees. A straightforward search of a three dimensional co-ordinate list involves $O(N^3)$ operations In terms of c.p.u. time, the dominant term is the set up time for the tree rather than the search time. The set up time is O(N) and even for moderate N (100-200), the binary tree approach shows dramatic gains. One search through the nodes checks for duplications. A similar search through the elements is used to check inter-element connectivities.



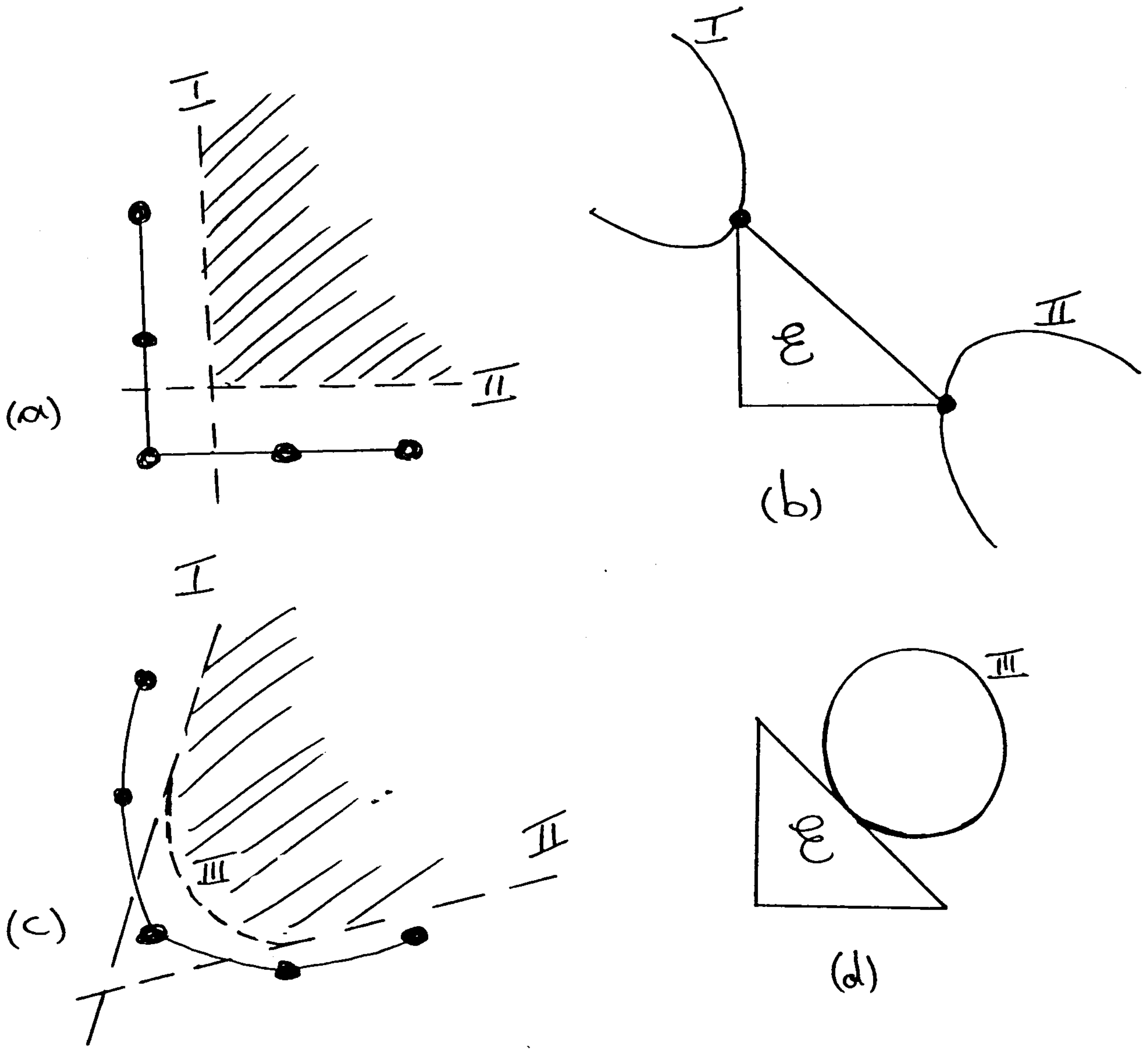
When considering curved grids various possible invalid elements have to be checked. Tests for (a) and (b) were developed by R.B. Simpson (see these proceedings). An element-wise search is adopted for (c). A test on the local transformation is necessary to avoid (d) - nonconvex elements and, (e) or (f) - overspill. In both these latter cases, the interior of the reference element E does not correspond to the region enclosed within the boundary $\partial E$ of the physical element E.


E
y
x
$T_0$
T
(0,1)
$\underset{\sim}{a} + \begin{pmatrix}1/2\\1/2\end{pmatrix}$
$\underset{\sim}{b} + \begin{pmatrix}0\\1/2\end{pmatrix}$
$\mathcal{E}_0$
(0,0)
(1,0)
$\underset{\sim}{c} + \begin{pmatrix}1/2\\0\end{pmatrix}$
Y
X
$\mathcal{E}$
$T_1$
$\eta$
$\xi$

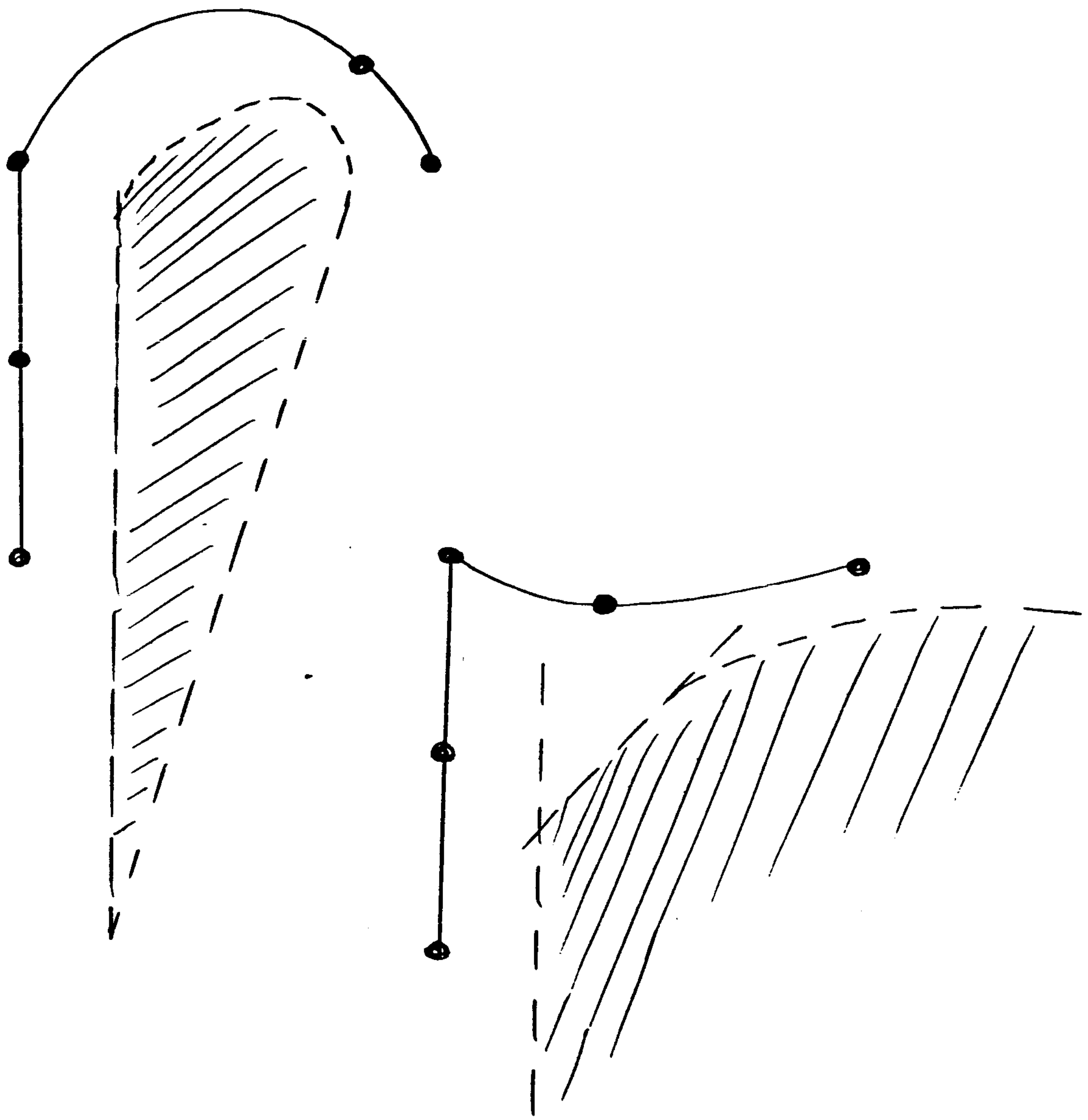
The general transformation $T: E_0 \rightarrow E$ is linear (bilinear for quadrilaterals) and forms the basis for the nonconvexity test. For isoparametric elements the Jacobian of $T_1$ is a polynomial in the perturbations $\underset{\sim}{a}$, $\underset{\sim}{b}$ and $\underset{\sim}{e}$ for the six-node triangle it is second degree in each of the perturbations.



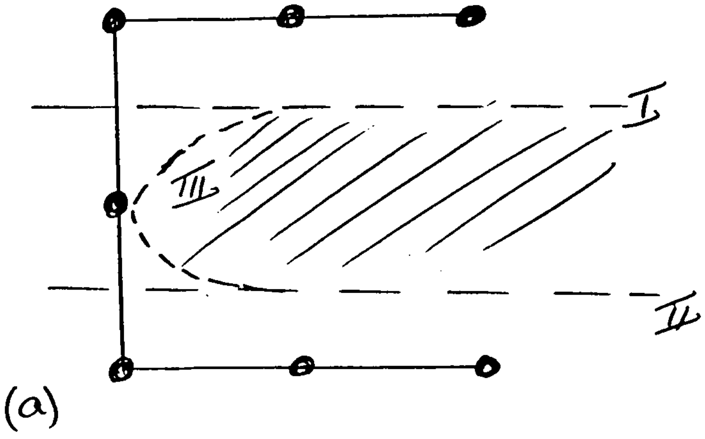
The validation of the local transformation requires the 1-1 correspondence of the simple closed curves $\partial \mathcal{E}$ and $\partial E$. Then the condition on the Jacobian of the form $J \gg 0$ is sufficient to ensure a valid element and an invertible transformation. The method adopted is to view the condition $J=0$ as defining a curve with exterior $J > 0$ that must contain the reference element . Thus (a) leads to a valid element, (b) and (c) are limiting cases and (d), (e), (f) and (g) are invalid.

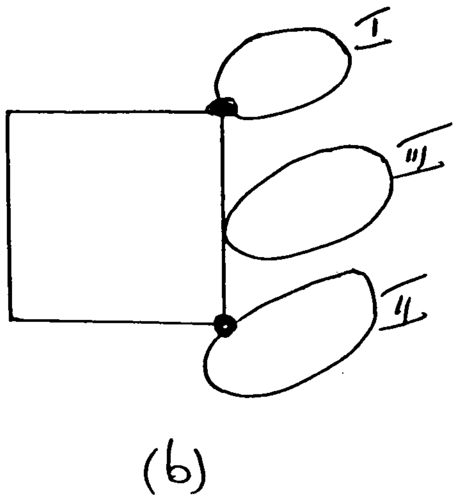


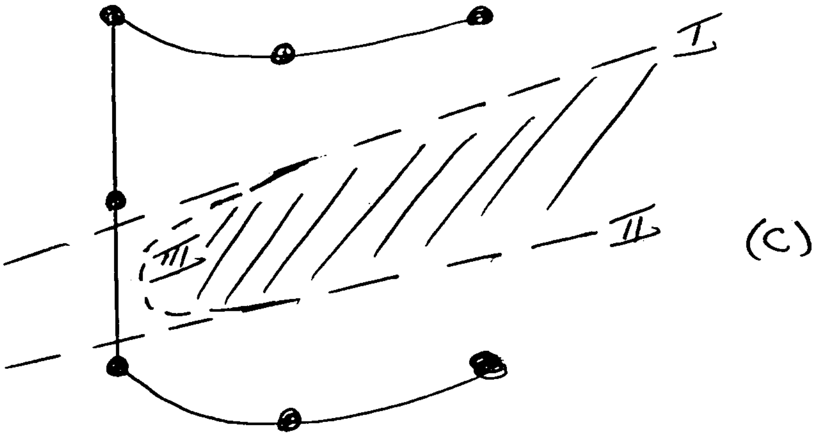
If two of the sides of a six-node triangle are fixed then the condition J > 0 can be viewed can be viewed as a restriction of the possible locations of the node on the free side. The boundary of the feasible region can be split into arcs, each arc arising from a limiting case of figure 5. If two sides are straight (a), the feasible region is a wedge with straight sides. The limiting cases corresponding to J=0 at a vertex always reduces to a linear condition if the sides are curved (c), then there is an additional non-linear arc corresponding to the tangential condition (d).



Examples of the feasible region for a side node of an element with two curved sides.







An analogous approach can be adopted for other forms of elements such as quadrilaterals. For the 8-node element the three conditions reduce to arcs; two linear and one nonlinear. These bound the feasible region for the position of one side node if all the others are assumed fixed. For elements with more than one side (or edge) node it is not possible to provide a simple geometric interpretation of the condition but it is still easy to represent it algebraically.

## References

R.B. Simpson "A two-dimensional mesh verification algorithm", these proceedings.

N.G. Brown and R. Wait "Verification and Inversion of isoparametric transformations" to appear.

A.R. Mitchell and R. Wait "The finite element method in partial differential equations", Wiley, London, 1977.

# Grid Generation for General Three-Dimensional Configurations

by

K. D. Lee, M. Huang, N. J. Yu and P. E. Rubbert
The Boeing Company
Seattle, Washington

## Abstract

The objective of the present study is to construct a suitable grid system for complex 3-D configurations such as a wing/body/nacelle shape for the solution of nonlinear transonic flow problems. Two approaches have been explored based on Thompson's body-fitted coordinate concept. The most general approach is to divide the computational domain into multiple rectangular blocks where the configuration itself is also represented by a set of blocks, whose structure follows the natural lines of the configuration. The block-structured grid system is adaptable to complex configurations and gives good grid quality near physical corners. However, it introduces algorithm issues for the flow solution concerning the treatment of nonanalytic grid block boundaries and nonstandard grid cells. These issues have been explored in relation to the grid generation. A more limited approach treats a wing/body configuration with only a single rectangular block in computational space. In this treatment the issues involving nonstandard cells are avoided, but other limitations on grid resolution appear. Both a linear and a nonlinear system of grid generation equations have been developed including methods of grid control. The linear method can generate grids of comparable quality with order-of-magnitude less cost. Its disadvantage is the greater possibility of ill-conditioned grids which, however, can be easily controlled in the block-structured grid system.

## Grid Generation Equations

1 Linear System

$$\vec{x}_{\xi\xi} + B\vec{x}_{\eta\eta} + C\vec{x}_{\zeta\zeta} + D\vec{x}_{\xi} + E\vec{x}_{\eta} + F\vec{x}_{\zeta} + G = 0$$

$\vec{x}$ = $(x, y, z)$

B to G: grid control
functions of $\xi$, $\eta$, and/or $\zeta$

2 Nonlinear System

$$A\left(\vec{x}_{\xi\xi} + \frac{P}{J^2A}\vec{x}_{\xi}\right) + B\left(\vec{x}_{\eta\eta} + \frac{Q}{J^2B}\vec{x}_{\eta}\right) + C\left(\vec{x}_{\zeta\zeta} + \frac{R}{J^2C}\vec{x}_{\zeta}\right) + 2\left(D\vec{x}_{\xi\eta} + E\vec{x}_{\xi\zeta} + F\vec{x}_{\eta\zeta}\right) = 0$$

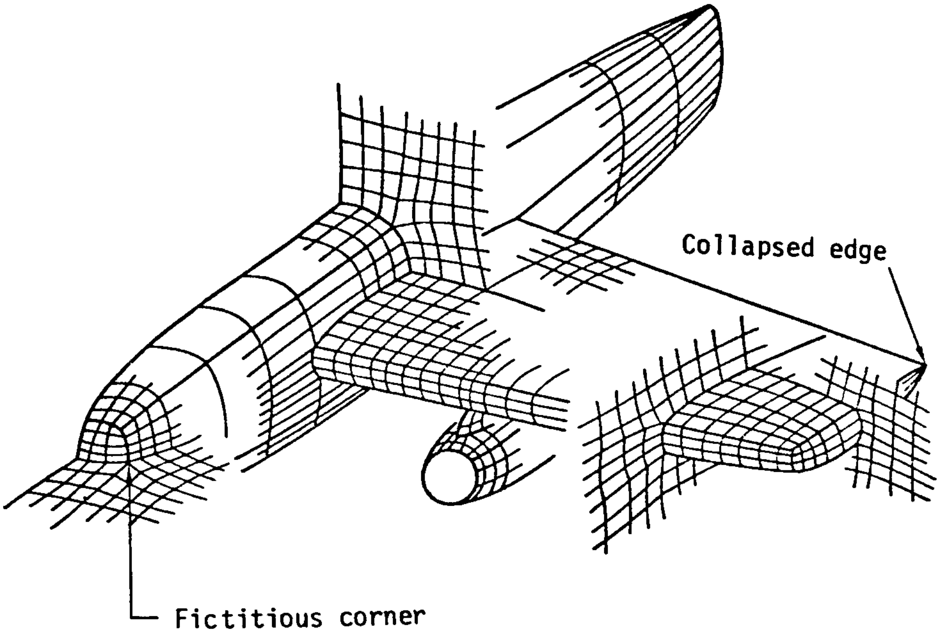
A to F: coupling terms
functions of x, y, and z

P, Q, R: grid control

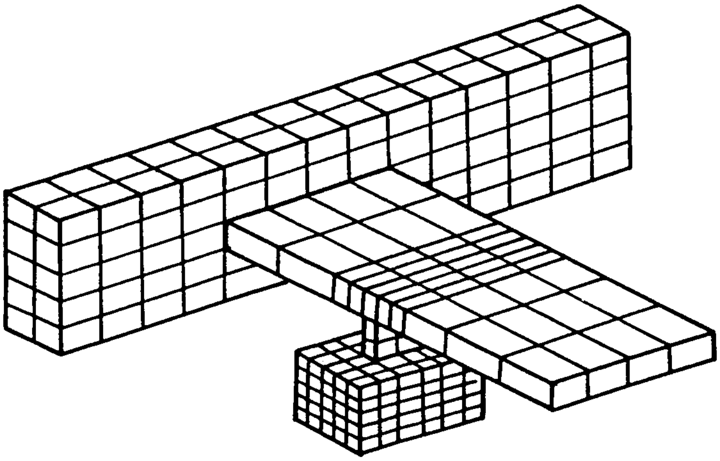
J = Jacobian of the transformation

Figure 1. Block structuring

This is a schematic illustration of a typical block structured grid about a wing/body/nacelle configuration. The multi-block grid obviously provides more desireable grid densities and eliminates the "lost corner." However, it introduces special points termed a "fictitious corner," a "collapsed edge," and a nonanalytic block boundary.



Physical space



Computational space

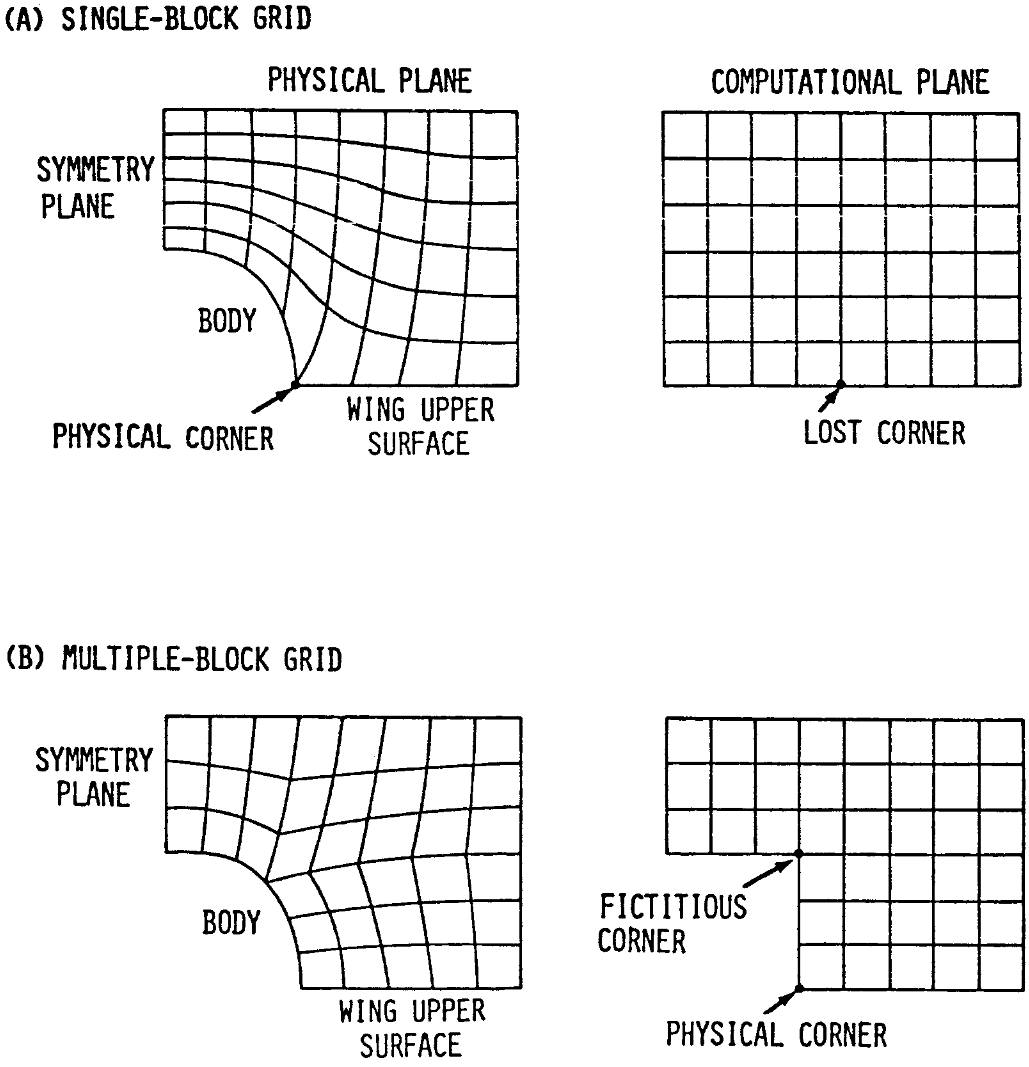
Figure 2. Comparison of grid structure

Lost corner - a physical corner transformed into a smooth point in the computational space

Fictitious corner - a smooth point transformed into a corner point in the computational space

Nonanalytic block boundary - grid lines across the block boundary are continuous but not smooth

Collapsed edge (3-D) - grid lines merge together in the physical space



(A) SINGLE-BLOCK GRID

(B) MULTIPLE-BLOCK GRID

Figure 3. Block-structured grid generation process

After defining the overall block structure, a one-dimensional grid generation along the block perimeters produces a perimeter discretization. This provides boundary conditions for a subsequent two-dimensional grid generation producing grids covering the block surfaces. These in turn serve as boundary conditions to produce three-dimensional volume grids filling each block.

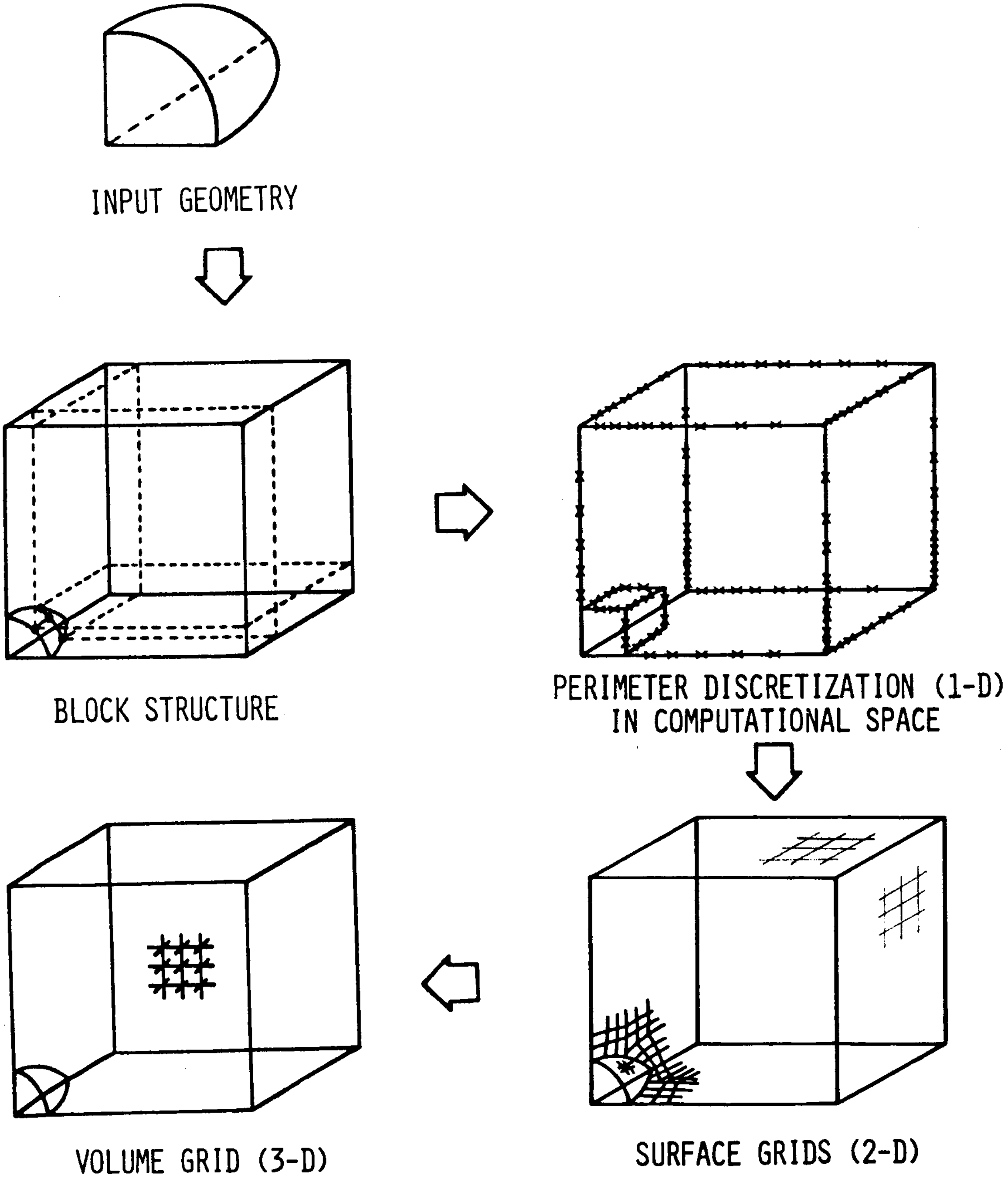
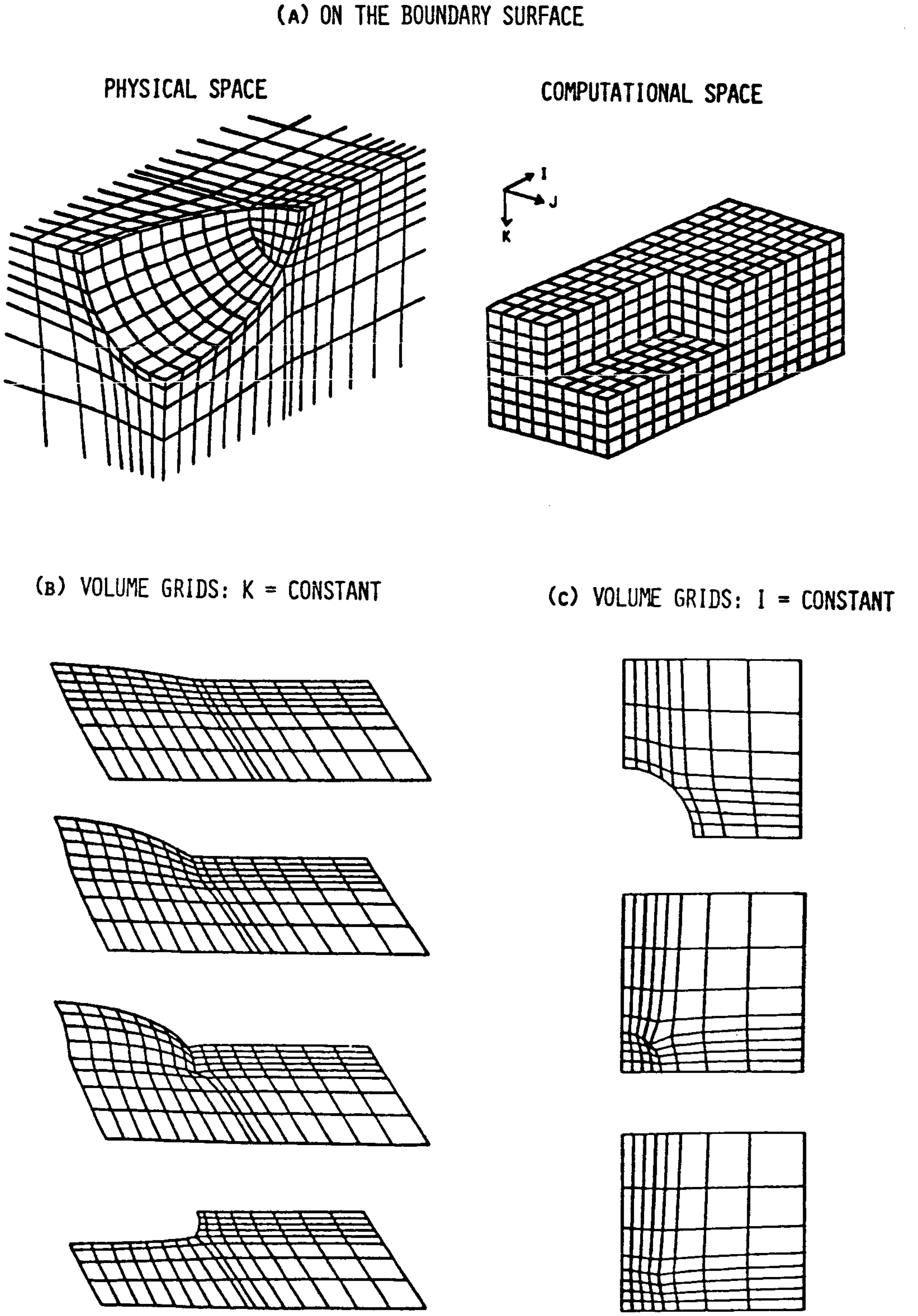


Figure 4. Block-structure grid for an ellipsoid

This example shows the grid around an ellipsoid which has been transformed to a cube in computational space. Fictitious corners can be seen.



(A) ON THE BOUNDARY SURFACE

PHYSICAL SPACE

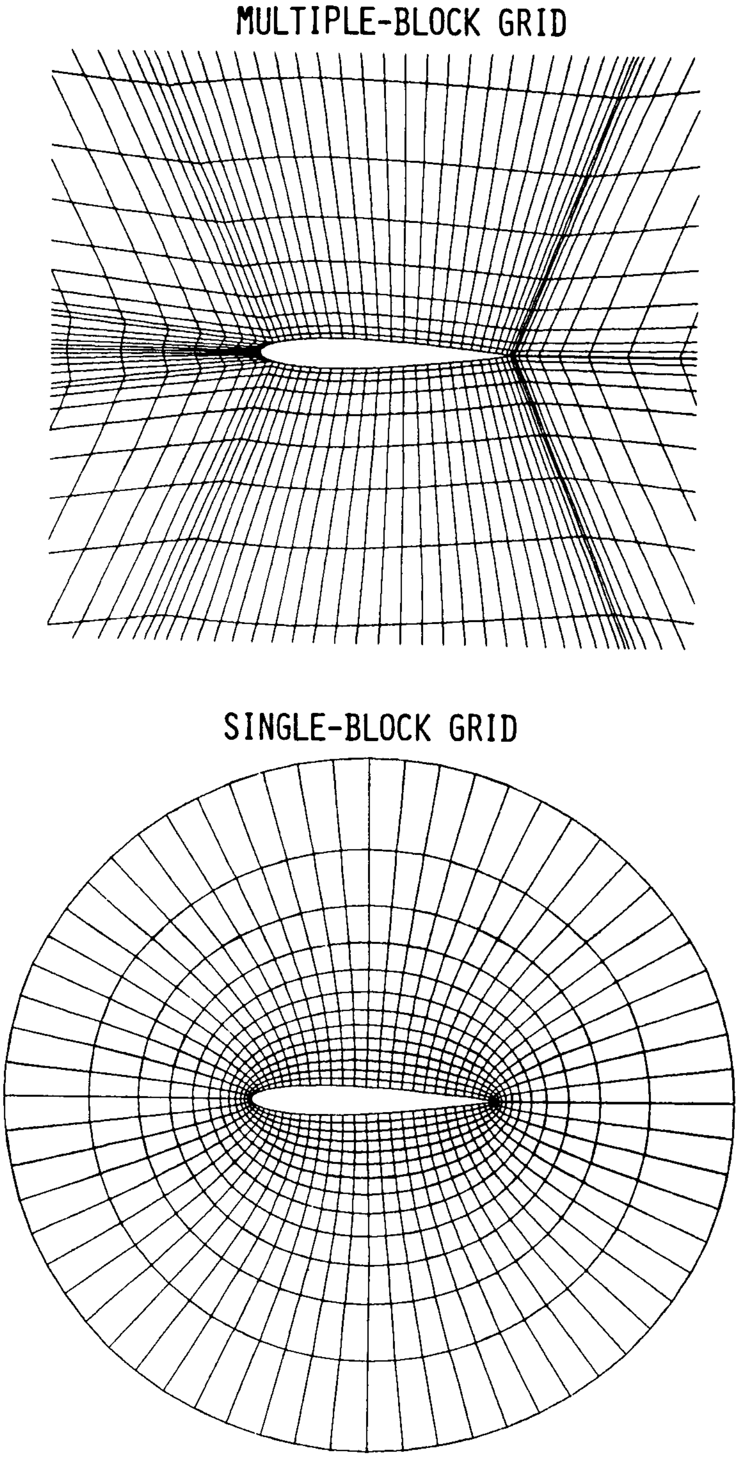
COMPUTATIONAL SPACE

(B) VOLUME GRIDS: K = CONSTANT

(C) VOLUME GRIDS: I = CONSTANT

Figure 6. Comparison of grids near an airfoil

The use of multi-block grid is considered for an airfoil. Compared to the ring-type grid, the multi-block grid seems to be overly complex. Its advantage is in its adaptability to more complex geometry.



MULTIPLE-BLOCK GRID

SINGLE-BLOCK GRID

Figure 5. Algorithm compatability study

The effect of the grid structure and the special points on the flow solution is explored by solving the potential flow over a cylinder. Cell-oriented flux formulation is used to treat the algorithm issues. Surprisingly, all the grid systems yield good resolution. Accuracy depends on the cell size rather than the grid structure at the special points.

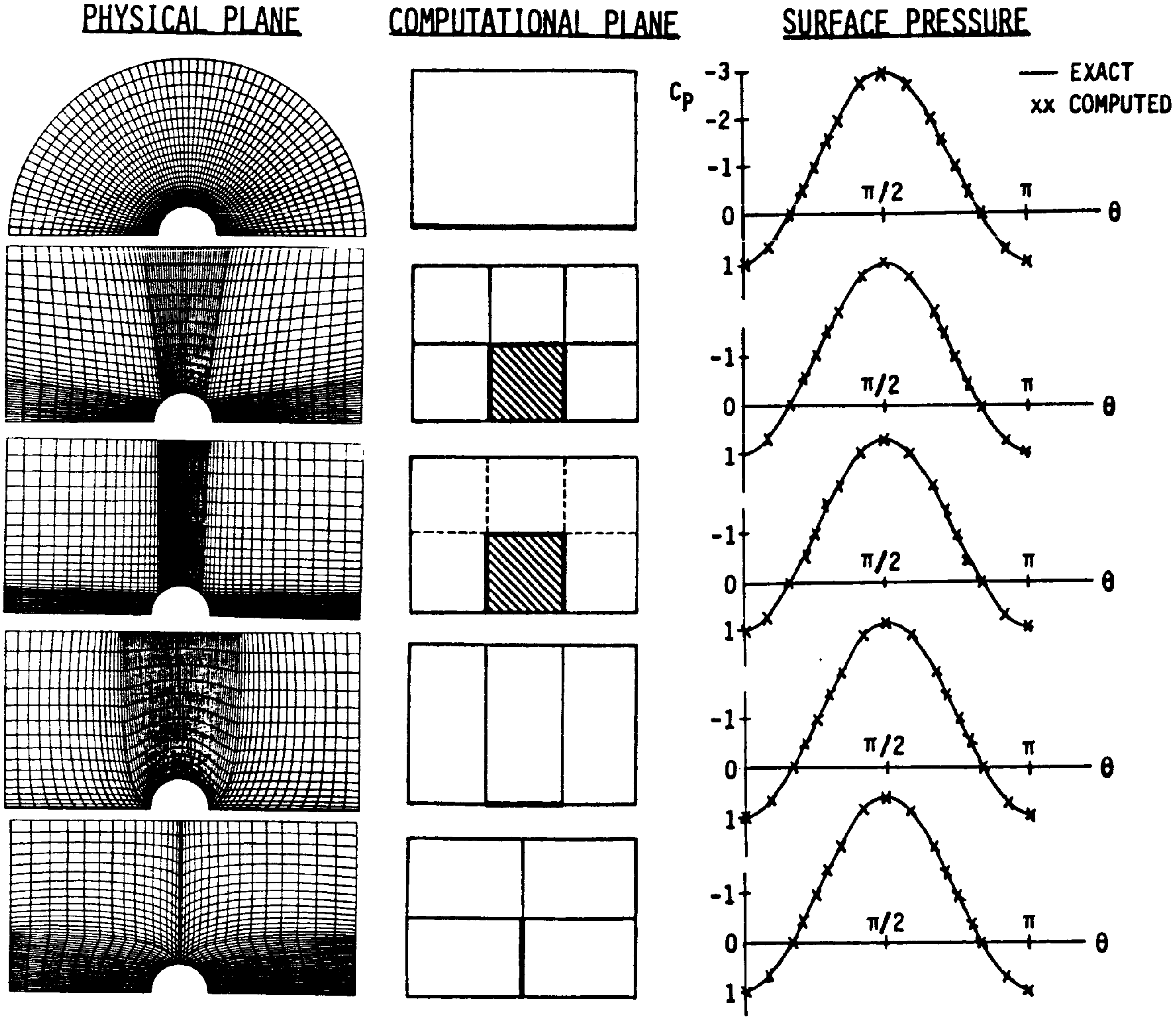


Figure 7. Airfoil study

The ability to produce accurate solutions using the multi-block grid is demonstrated in subsonic and transonic regions. Compared to the results from the ring-type single-block grid, remarkable accuracy was obtained even when the fictitious corner is located in supersonic regions. All the flow and metric quantities are defined at the center of each cell and the artificial density method is adopted for the density retardation in supersonic region.

PRESSURE DISTRIBUTION (NACA 0012 AIRFOIL)

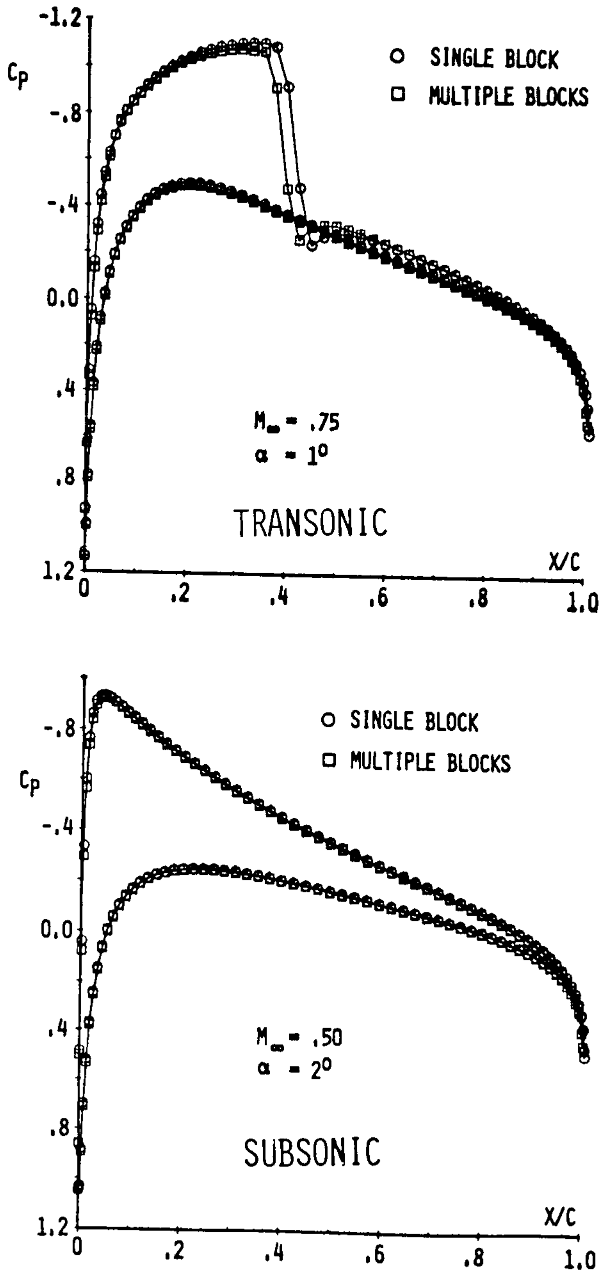


Figure 8. Surface grid for a wing/body
(single-block structure)

The use of the C-type grid provides smooth grid distribution near the wing leading edge. The body surface line on the symmetry plane coincides with a grid line which consists of lost corners. One concern is grid quality at the wing tip.

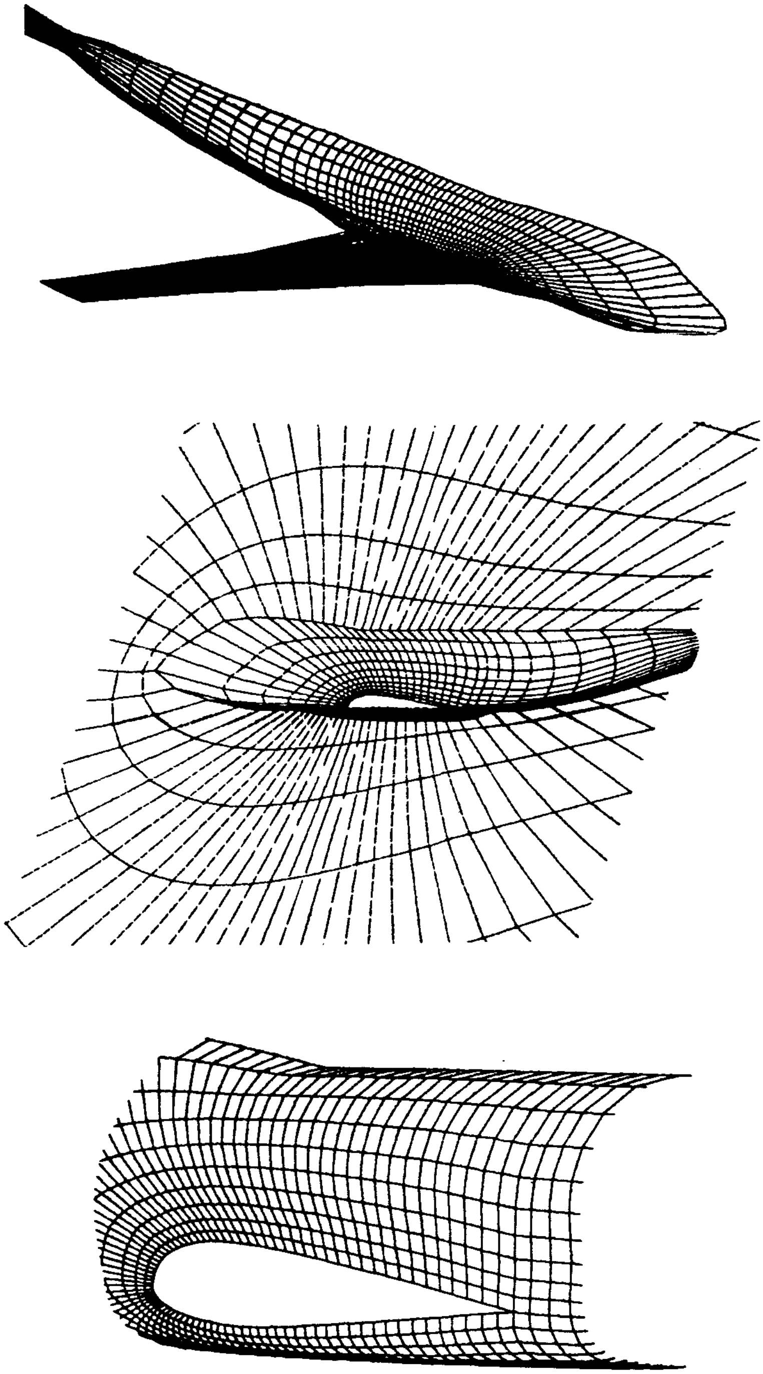


Figure 9. Surface grid for a wing/body
(multi-block structure)

The use of a multi-block grid eliminates the lost corners in the single-block grid of figure 8 and improves the grid quality near the wing tip, while producing the fictitious corners and nonanalytic block boundaries. Its ability to extend to more complex configurations is obvious.

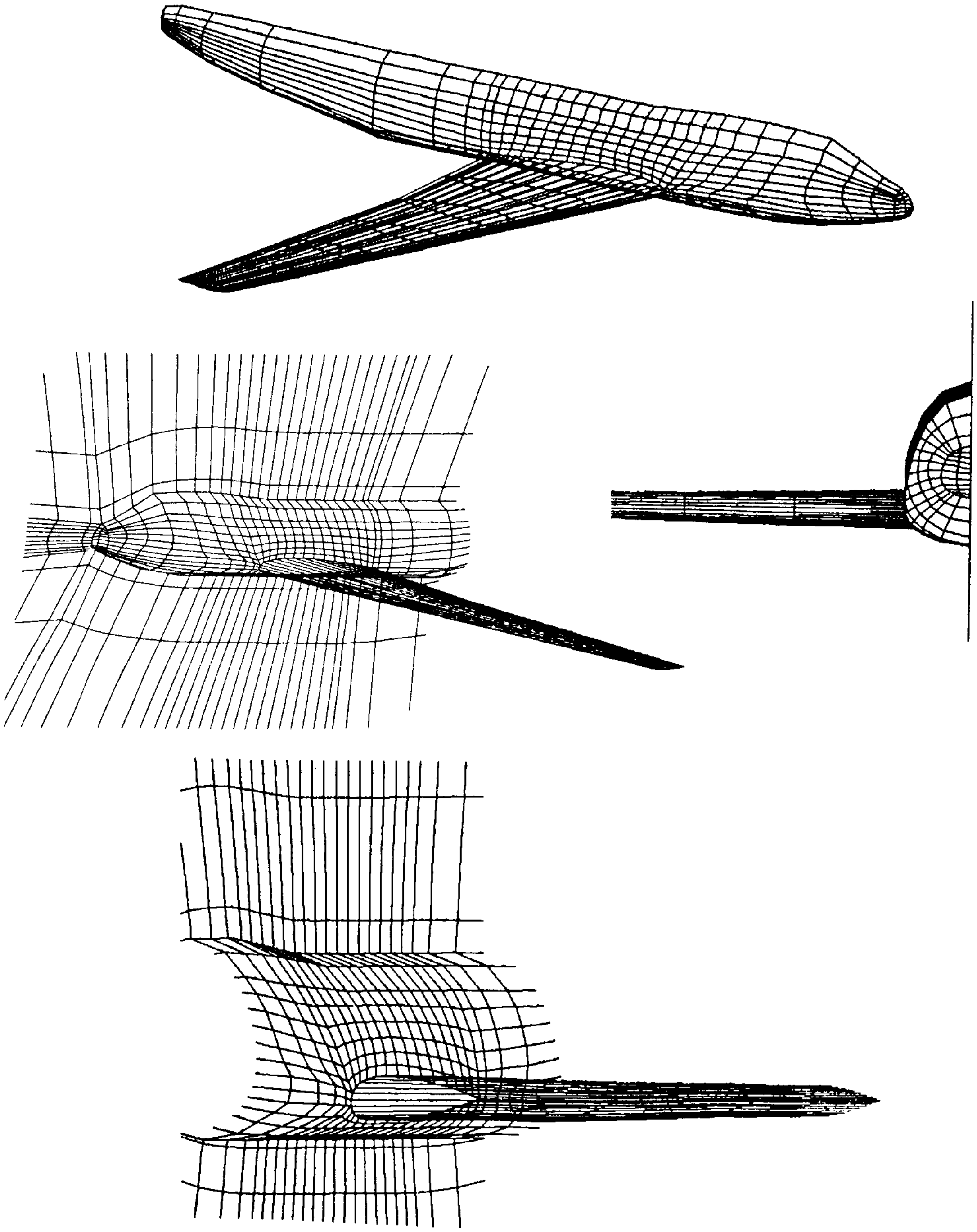
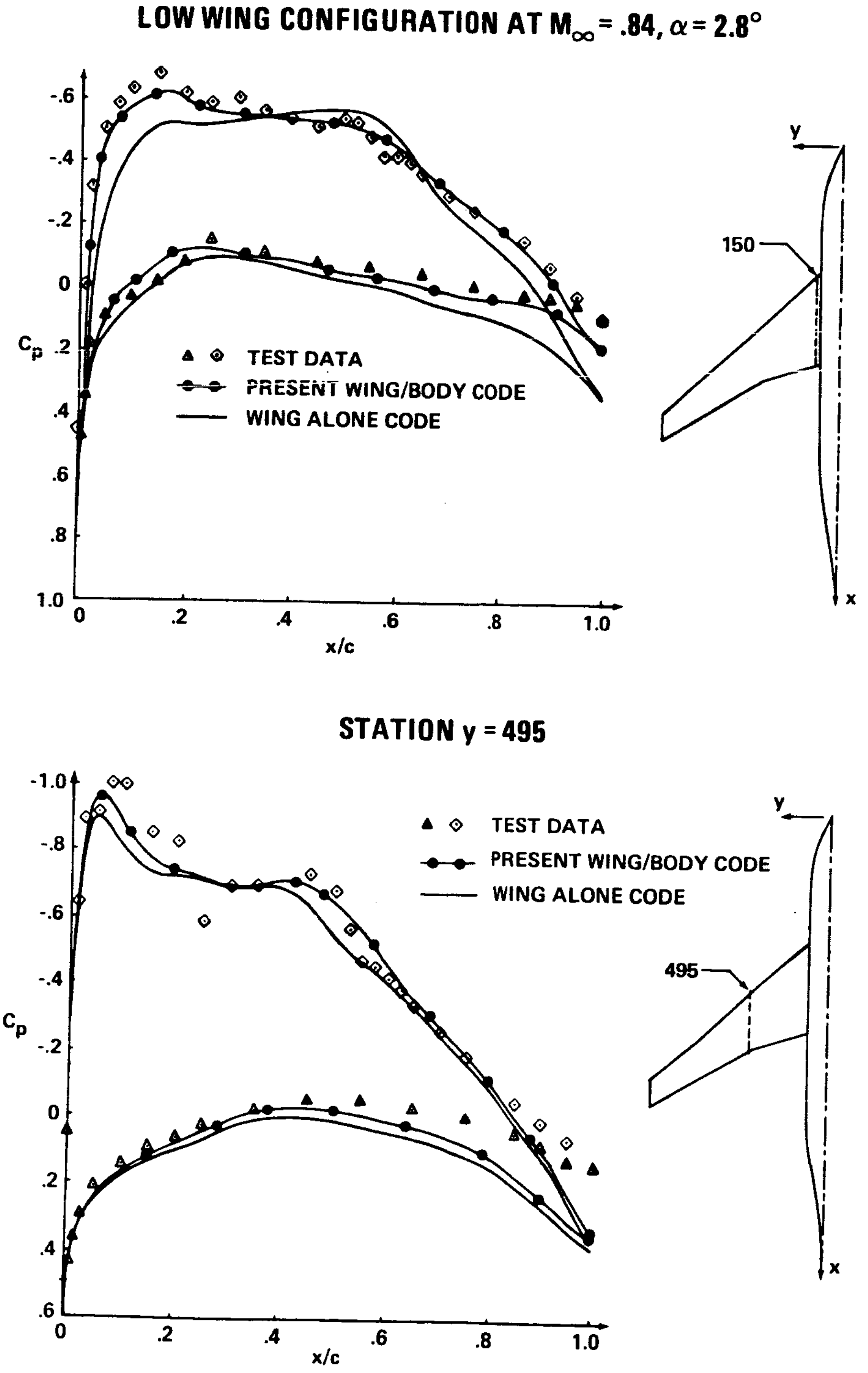


Figure 10. 3-D flow solution

A transonic solution for a wing/body combination is obtained using the single-block grid and compared to the experimental results. The use of body-fitted grid system improves the accuracy near the wing/body junction.

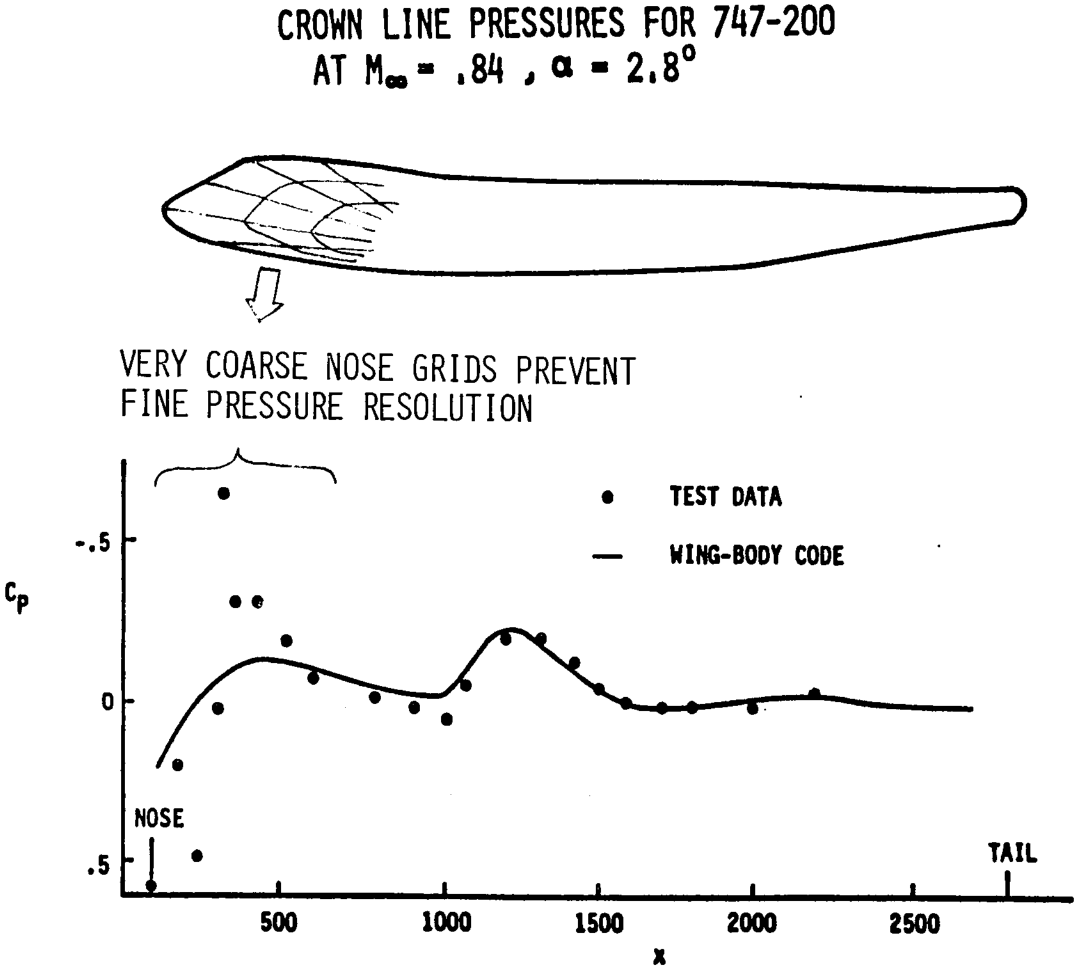


LOW WING CONFIGURATION AT $M_\infty = .84, \alpha = 2.8°$

STATION y = 495

Figure 11. 3-D flow solution

The body-fitted grid system can produce quite accurate pressure distribution even on the body surface. Very coarse nose grid distribution prevents fine pressure resolution in that region.



CROWN LINE PRESSURES FOR 747-200
AT $M_\infty = .84$, $\alpha = 2.8^\circ$

VERY COARSE NOSE GRIDS PREVENT FINE PRESSURE RESOLUTION

WORKSHOP ON GRID GENERATION
NASA Langley Research Center
October 6-7, 1980

# BOUNDARY-FITTED COORDINATE SYSTEMS FOR ARBITRARY COMPUTATIONAL REGIONS

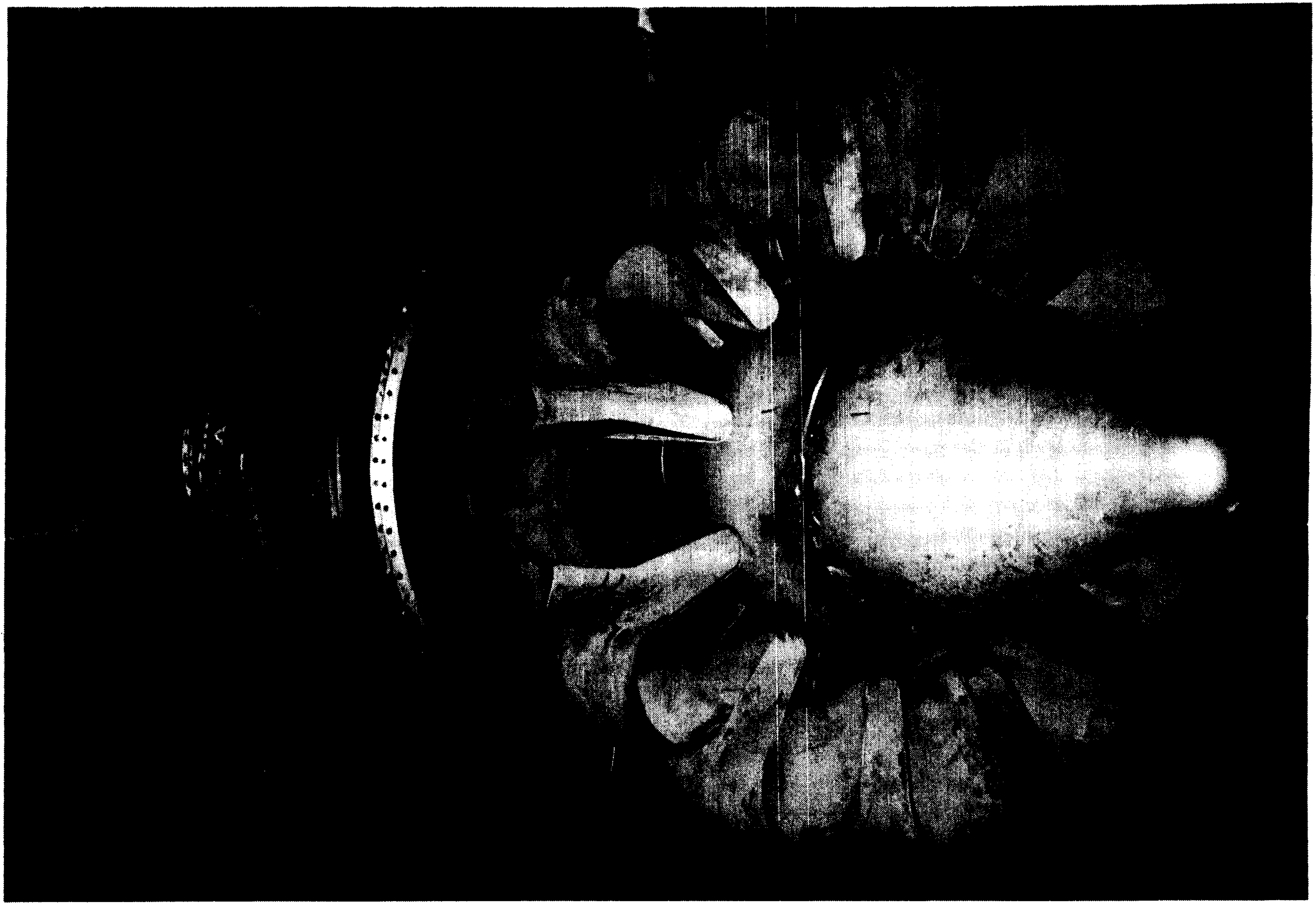
By Edward J. Kowalski
Boeing Military Airplane Company
Advanced Airplane Branch
Seattle, Washington 98124

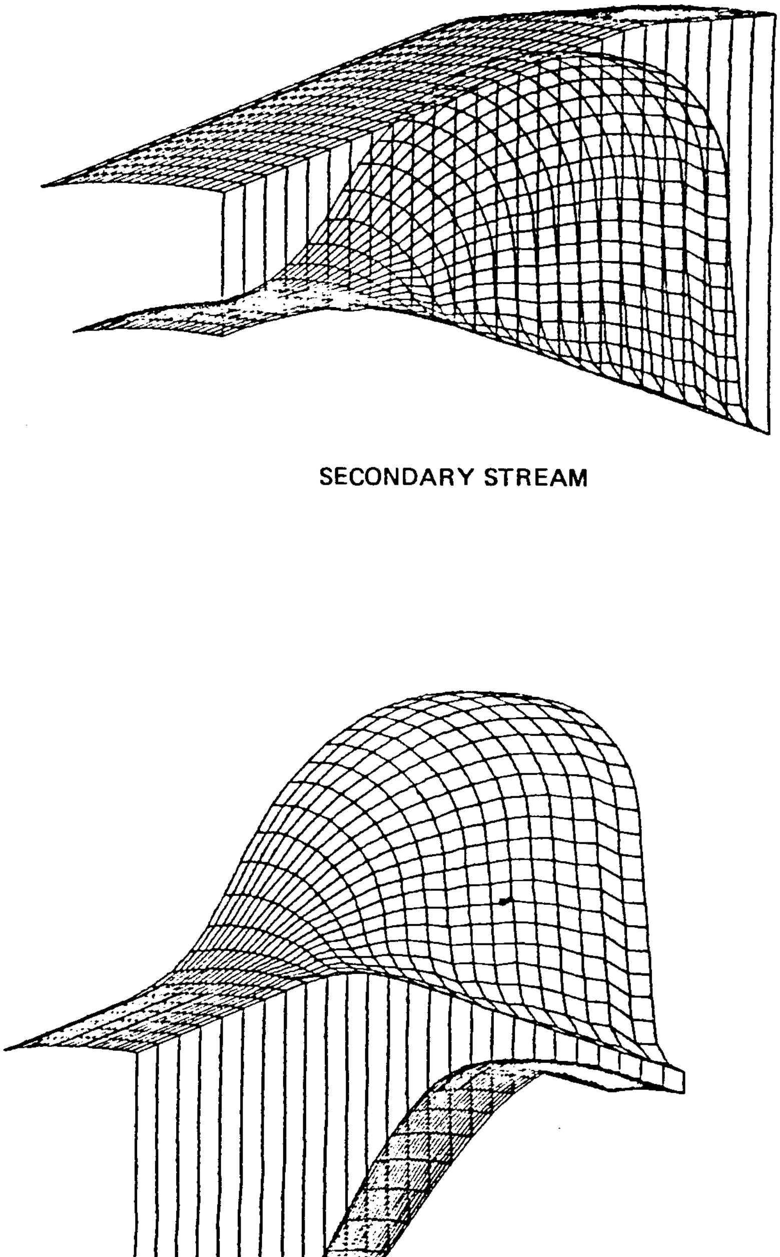
A computational region of arbitrary cross section presents a significant problem in the generation of a mesh. Simple orthogonal meshes are difficult to use because the mesh points do not naturally fall on the region's boundaries. Differencing and interpolation schemes become complex and cumbersome, and it is difficult to extend these schemes to higher order because of the complex logic required. Higher order schemes are desirable as they allow calculation of a flow to a given level of accuracy with a lower mesh density and hence less storage than a lower order scheme. High accuracy solutions are possible for a region of arbitrary cross-section when a boundary-fitted computational mesh is employed. A boundary-fitted mesh is defined as a mesh in which the boundary (i.e., a duct wall) is coincident with the mesh points that are used for finite difference expressions at, and adjacent to, the boundary. Interpolation is not required, and extension to higher order differencing is straightforward. This is a significant benefit when the boundary conditions have a dominant influence on the solution.

This paper will discuss the application of Smith and Wiegel's method for generating boundary fitted coordinate systems (discussed in their AIAA-80-0192 paper entitled, "Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Simulation") for two practical flow problems characterized by complex surface geometry:

- radial mixer lobe
- subsonic inlet designed for high angle-of-attack capability

# FULL SCALE FORCED MIXER

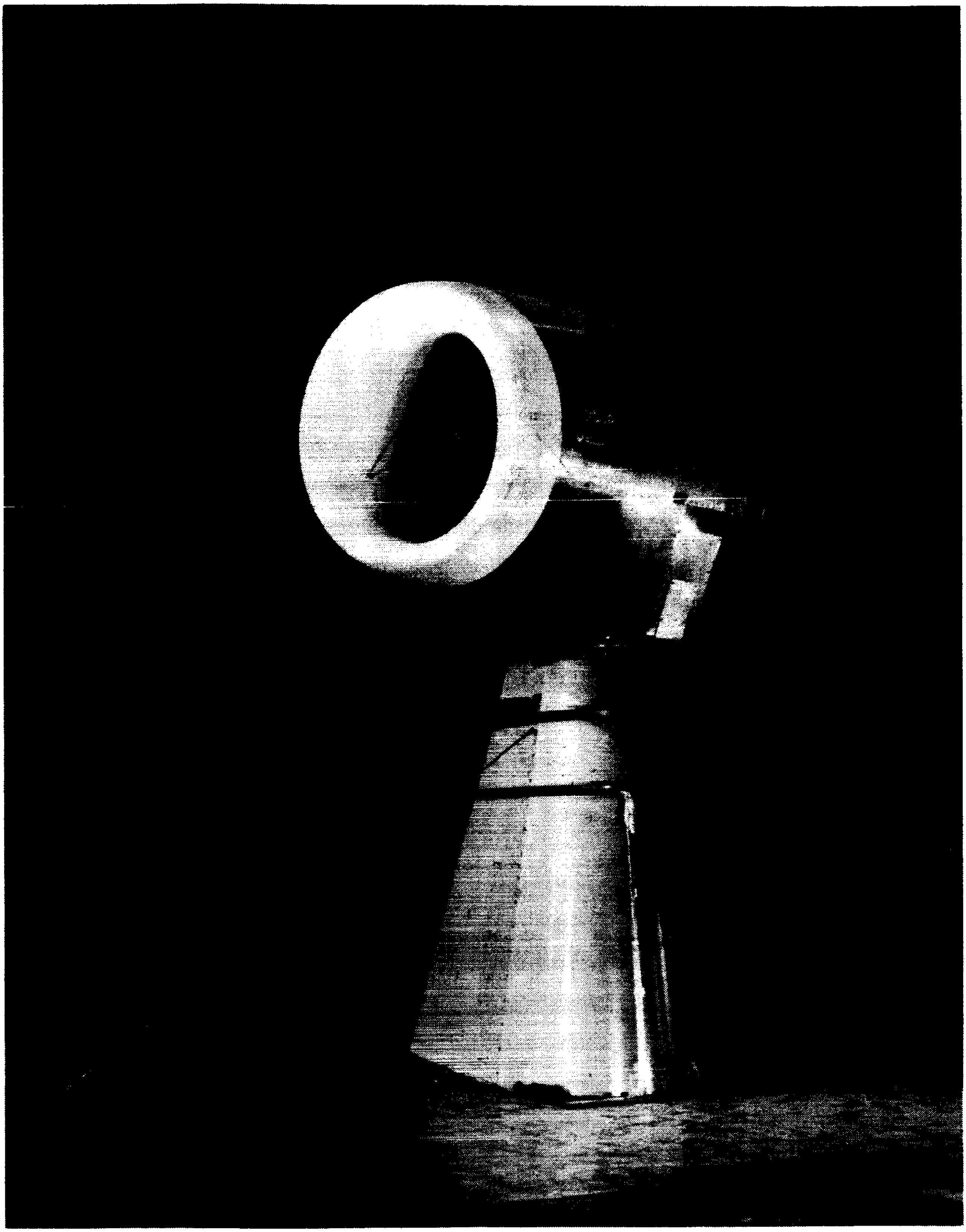


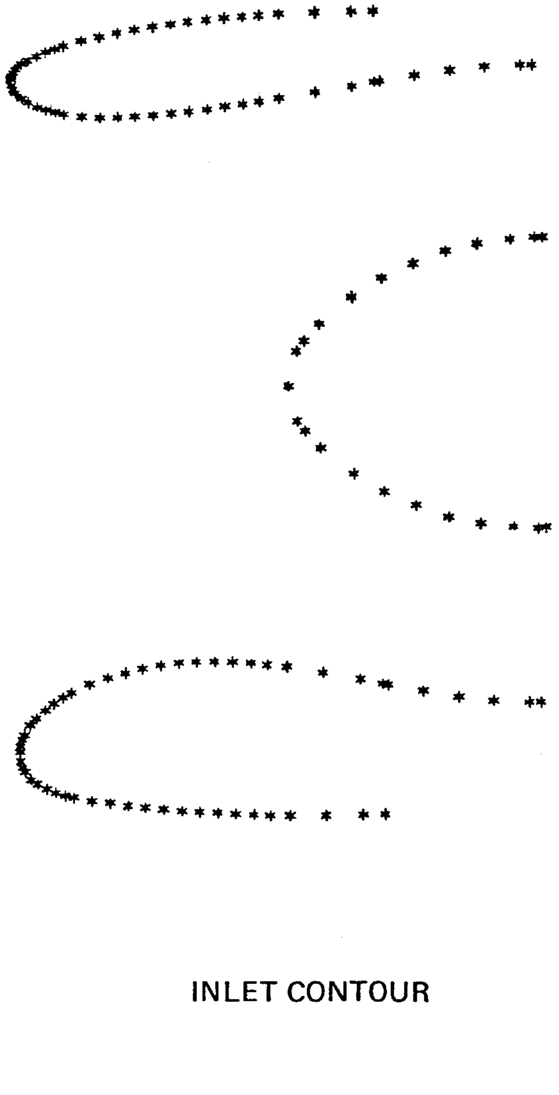


SECONDARY STREAM

PRIMARY STREAM

*Radial Mixer Lobe.*





INLET CONTOUR

In the method of Smith and Wiegel, two disconnected boundaries are defined and an explicit functional relation is used to establish the transformation between the physical domain and the computational domain.

The physical domain is defined by a cartesian coordinate system; the computational domain is defined with the variables $\xi$ , $\eta$ and $\zeta$ with the values:

$$0 \le \xi \le 1$$
$$0 \le \eta \le 1$$
$$0 \le \zeta \le 1$$

Two possible connecting functions are suggested: linear and a cubic parametric polynomial. The following cubic polynomial equation was used to generate meshes for both the lobe mixer and the subsonic inlet:

$$x = X_1(\xi,\zeta)f_1(\eta) + X_2(\xi,\zeta)f_2(\eta) + \frac{dX_1}{d\eta}(\xi,\zeta)f_3(\eta) + \frac{dX_2}{d\eta}(\xi,\zeta)f_4(\eta),$$

$$y = Y_1(\xi,\zeta)f_1(\eta) + Y_2(\xi,\zeta)f_2(\eta) + \frac{dY_1}{d\eta}(\xi,\zeta)f_3(\eta) + \frac{dY_2}{d\eta}(\xi,\zeta)f_4(\eta),$$

$$z = Z_1(\xi,\zeta)f_1(\eta) + Z_2(\xi,\zeta)f_2(\eta) + \frac{dZ_1}{d\eta}(\xi,\zeta)f_3(\eta) + \frac{dZ_2}{d\eta}(\xi,\zeta)f_4(\eta),$$

where:

$X_\ell(\xi,\zeta), Y_\ell(\xi,\zeta), Z_\ell(\xi,\zeta)$, $\ell = 1,2$ are the boundary points in the physical domain

$\frac{dX_\ell}{d\eta}(\xi,\zeta)$, $\frac{dY_\ell}{d\eta}(\xi,\zeta)$ $\frac{dZ_\ell}{d\eta}(\xi,\zeta)$, $\ell = 1,2$ are the derivatives of the boundary points in the physical domain

$$f_1(\eta) = 2\eta^3 - 3\eta^2 + 1,$$
$$f_2(\eta) = -2\eta^3 + 3\eta^2,$$
$$f_3(\eta) = \eta^3 - 2\eta^2 + \eta,$$
$$f_4(\eta) = \eta^3 - \eta^2.$$

The cubic connecting function forces orthogonality at the boundaries of the physical domain by calculating the derivatives $\frac{dX_\ell}{d\eta}(\xi,\zeta)$ $\frac{dY_\ell}{d\eta}(\xi,\zeta)$ and $\frac{dZ_\ell}{d\eta}(\xi,\zeta)$ from the cross product of the tangential derivatives and then dividing by the magnitude of the normal vector.

Four extensions of the Smith and Wiegel method were necessary in order to successfully apply their technique to the mixer lobe and subsonic inlet.

First, because of the nature of the mixer and inlet geometries, points defining the boundaries had to be positioned using a geometric progression.

$$S = a + ar + ar^2 + \dots + ar^{N-1}$$
$$= \frac{a(1-r^N)}{1-r}$$

where:

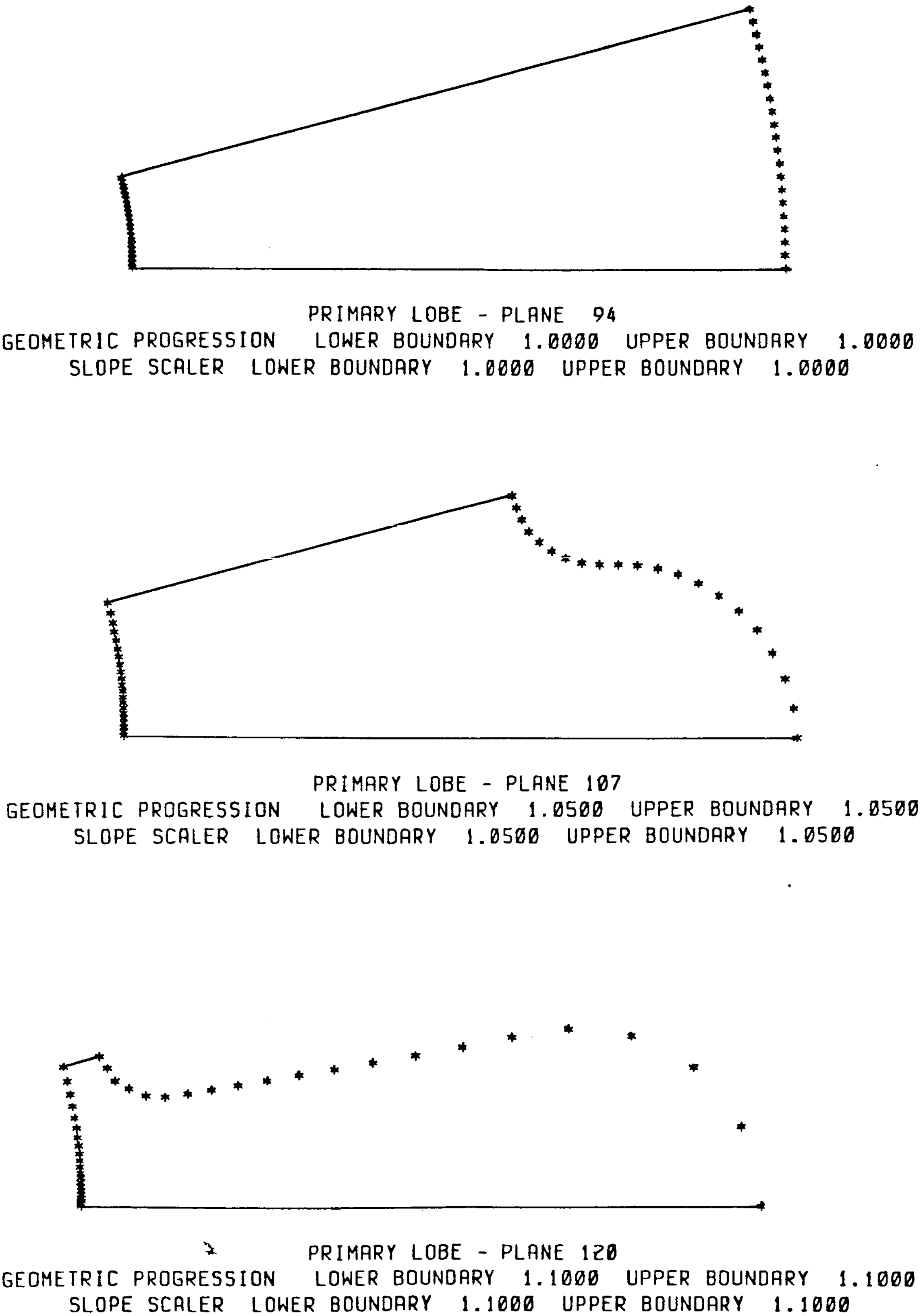
$S$ = the total length of the boundary

$a$ = first increment

$r$ = scale factor

$N$ = number of cells (one less the number of boundary points)

For the mixer, the scale factor $r$ was varied linearly from $r = 1$ at the mixer entrance plane (where the boundary is an arc) to $r = r_{max}$ at the mixer exit plane (where the boundary is highly distorted). This makes it possible to force the mesh to migrate to regions of interest without causing significant distortions in the mesh from plane to plane. The optimal distribution of mesh occurred when the upper and lower boundary mesh points were stretched in opposite directions.

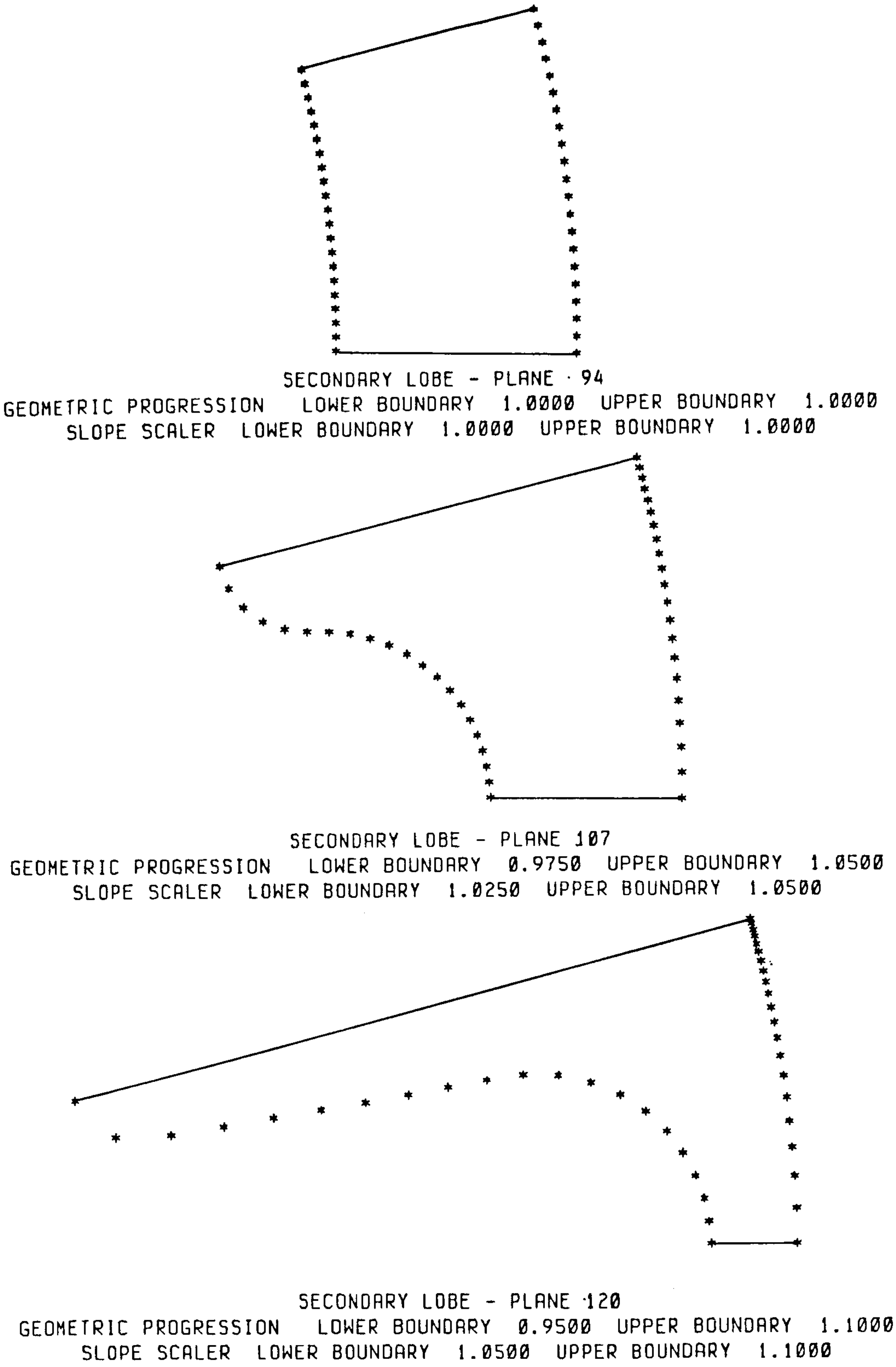


PRIMARY LOBE - PLANE 94
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000
SLOPE SCALER LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000

PRIMARY LOBE - PLANE 107
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.0500
SLOPE SCALER LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.0500

PRIMARY LOBE - PLANE 120
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000
SLOPE SCALER LOWER BOUNDARY 1.1000 UPPER BOUNDARY 1.1000

*Geometric Progression of Boundary Points for the Primary Stream.*



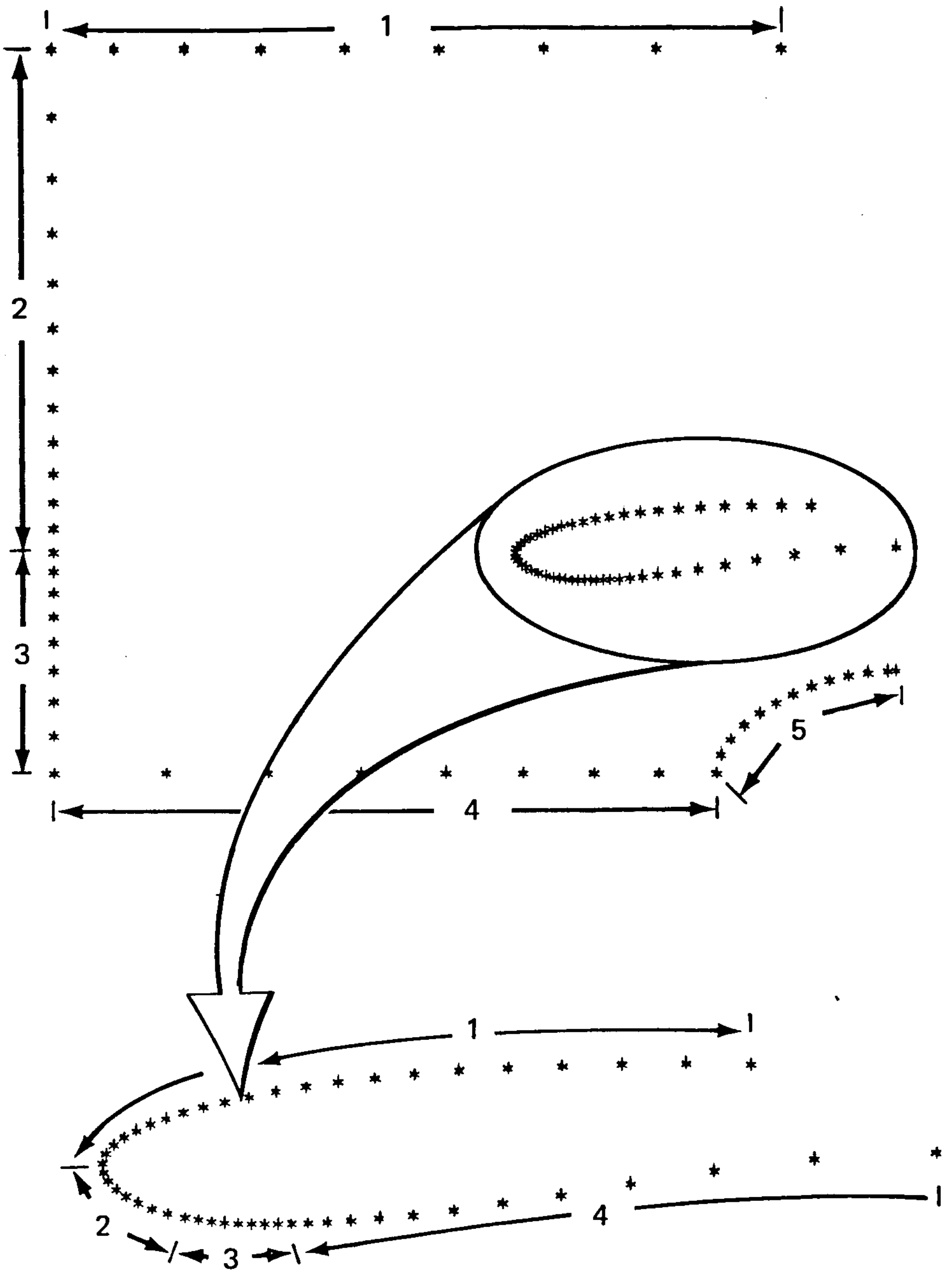
SECONDARY LOBE - PLANE 94
GEOMETRIC PROGRESSION LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000
SLOPE SCALER LOWER BOUNDARY 1.0000 UPPER BOUNDARY 1.0000

SECONDARY LOBE - PLANE 107
GEOMETRIC PROGRESSION LOWER BOUNDARY 0.9750 UPPER BOUNDARY 1.0500
SLOPE SCALER LOWER BOUNDARY 1.0250 UPPER BOUNDARY 1.0500

SECONDARY LOBE - PLANE 120
GEOMETRIC PROGRESSION LOWER BOUNDARY 0.9500 UPPER BOUNDARY 1.1000
SLOPE SCALER LOWER BOUNDARY 1.0500 UPPER BOUNDARY 1.1000

*Geometric Progression for Boundary Points for the Secondary Stream.*

The inlet has certain regions (hilite, throat, etc.) which require a fine computational mesh to insure a detailed analysis. For this reason, four regions along each inlet contour and five regions along the boundary of the analysis domain required individual geometric progressions. The scale factor, $\Gamma$ and the number of cells, N, of each regions must be chosen to insure a smooth progression in cell length along each of the boundaries.

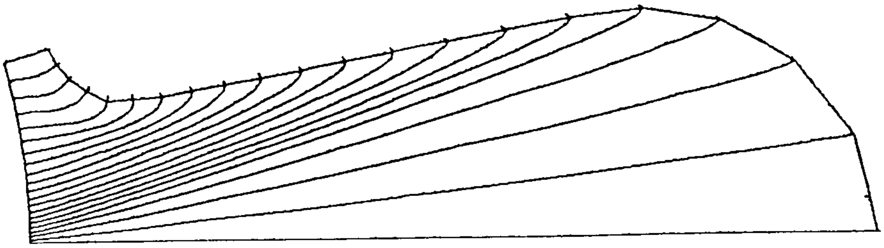


GEOMETRIC PROGRESSION REGIONS ALONG THE INLET CONTOUR AND THE ANALYSIS BOUNDARY

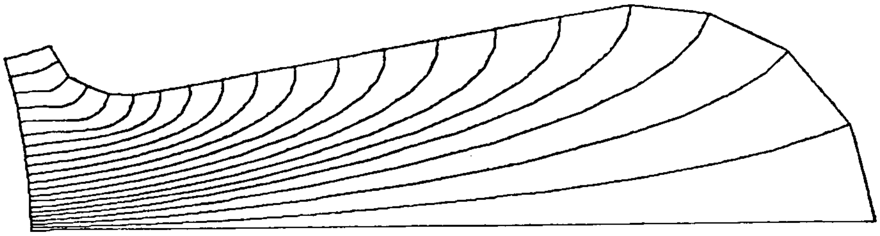
The second extension uses a ramping function to regulate the dependence of the connecting function on the boundary slope. This connecting function is an explicit functional relation used to establish the transformation between the physical domain and the computational domain.

For the mixer lobe, this dependence was regulated to redistribute the internal mesh points and reduce mesh skewness.

In the case of the subsonic inlet, it was found that a constant value for each plane was sufficient to insure against mesh line cross-over.



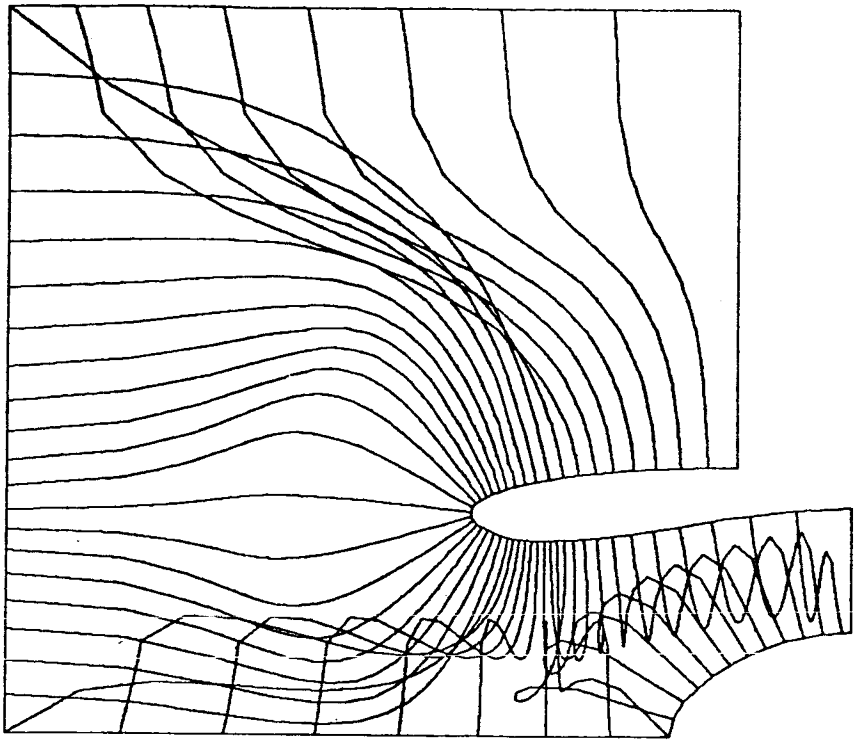
WITHOUT RAMPING FUNCTION



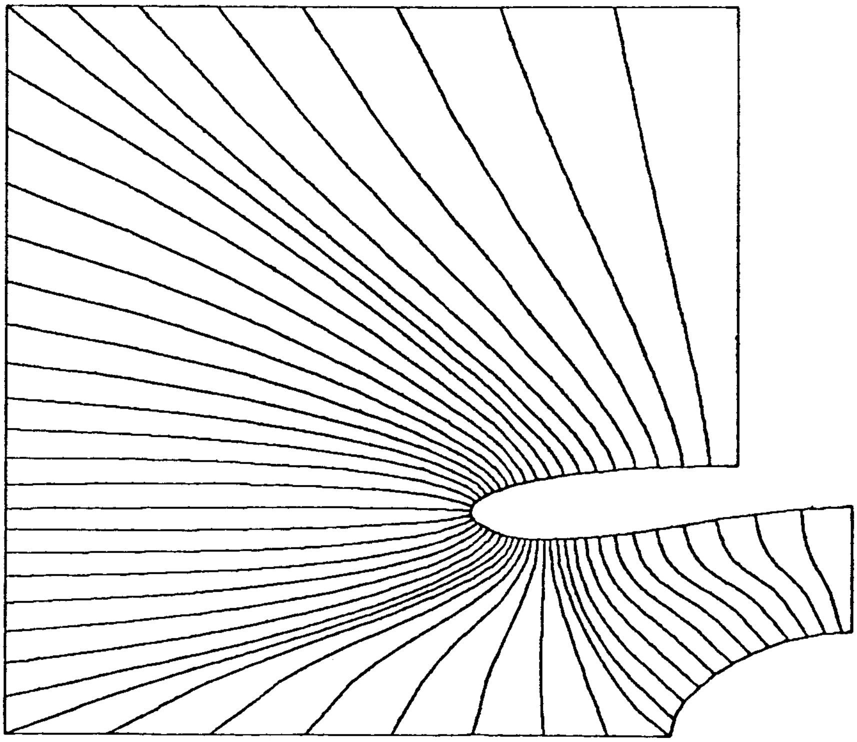
WITH RAMPING FUNCTION

*Figure 4. Connecting Function Dependency on Boundary Slope.*

WITHOUT RAMPING FUNCTION



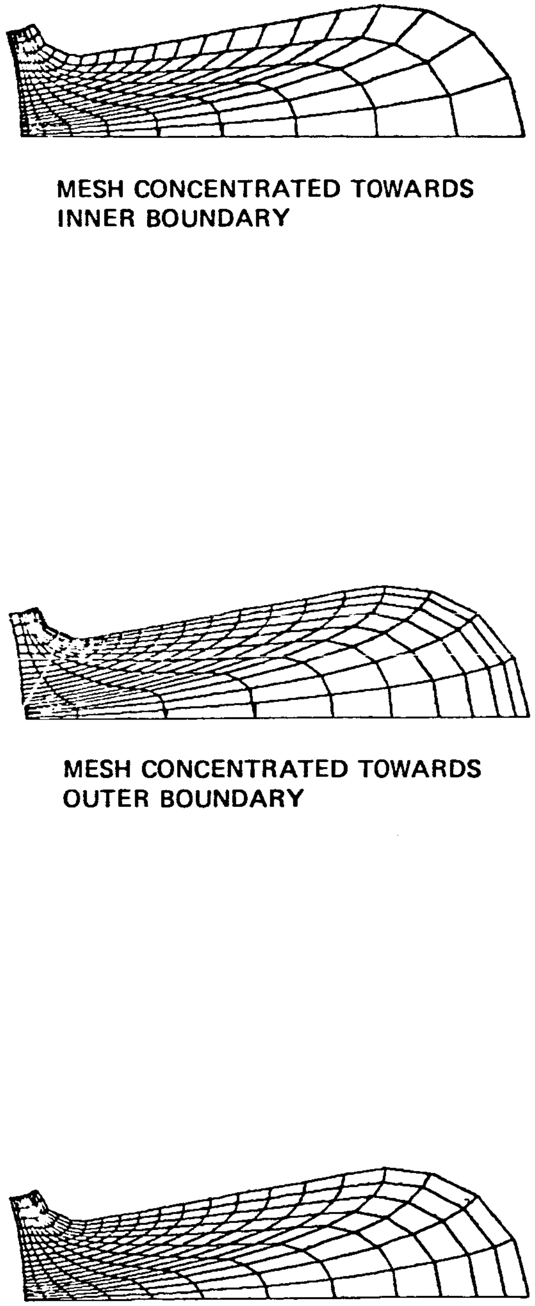
WITH RAMPING FUNCTION



CONNECTING FUNCTION DEPENDENCY ON BOUNDARY SLOPE

The third extension utilizes the concentration function suggested by Smith and Wiegel, but uses it to force the mesh in the direction of both boundaries of the mixer lobe. More mesh was then needed to be linearly added to fill the void created by this mesh concentration.

The inlet only required the mesh to be forced towards the inlet contour. A concentrated mesh was assumed unnecessary along the spinner boundary; it was felt that for a potential flow analysis the flow about the spinner would not propagate upstream and affect the solution at the regions of interest (hilite, throat, etc.). The mesh concentration for both the mixer and the inlet permits flow analysis within the boundary layers.

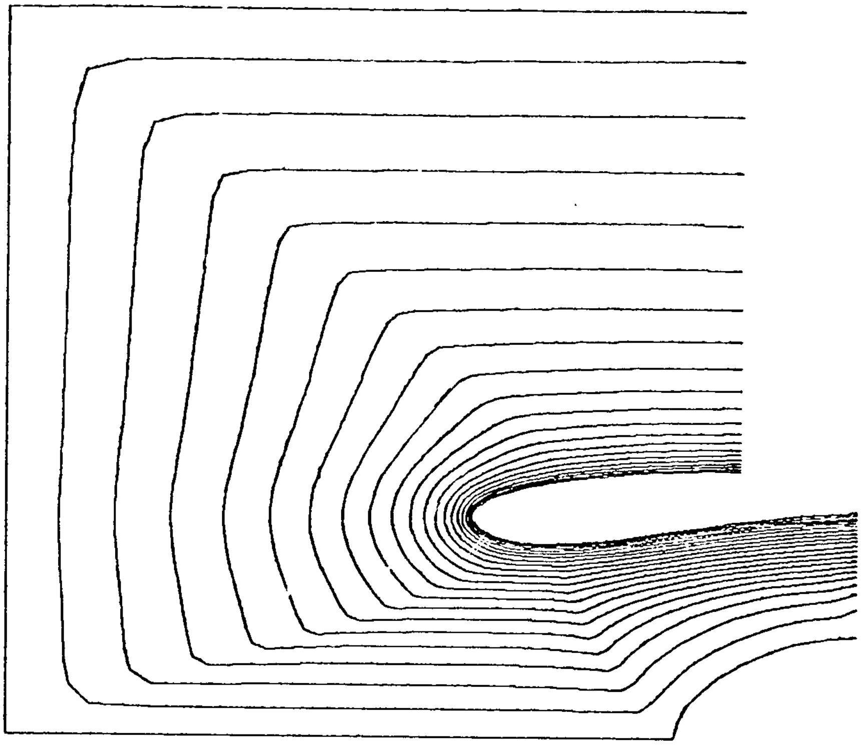


MESH CONCENTRATED TOWARDS
INNER BOUNDARY

MESH CONCENTRATED TOWARDS
OUTER BOUNDARY

MESH CONCENTRATED TOWARDS
BOTH BOUNDARIES

*Figure 5. Mesh Concentration.*



MESH CONCENTRATION

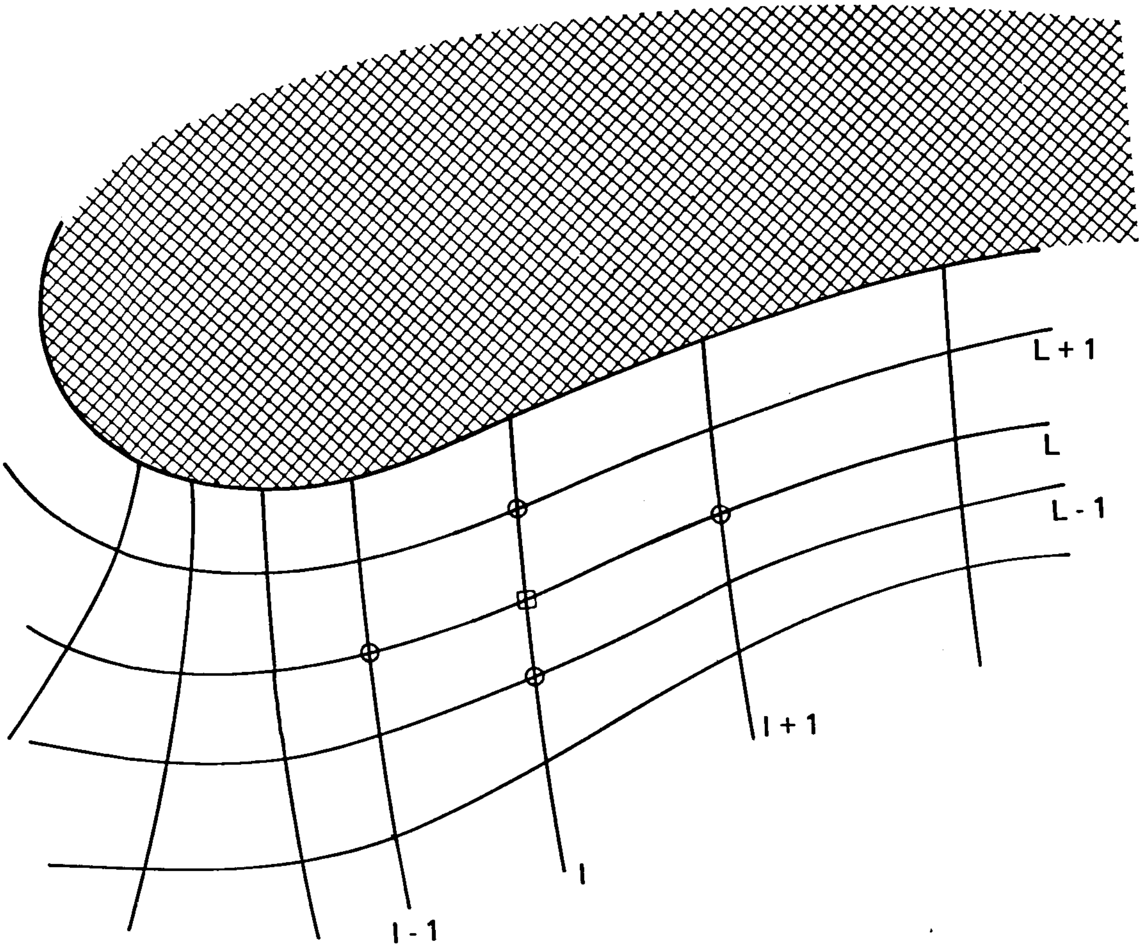
The fourth extension applies to the subsonic inlet only. It was necessary to produce a computational mesh which possessed a smooth progression of cell metrics and cell volumes in all directions to allow a solution process of a flow analyser to use the grid efficiently. The interior points of the computational mesh were "smoothed" by a multiple application of a five point diffusion operator:

$$X(L,I)_{new} = \alpha \left\{ X(L-1,I) + X(L+1,I) + X(L,I-1) + X(L,I+1) - 4 * X(L,I)_{old} \right\}$$

$$Y(L,I)_{new} = \alpha \left\{ Y(L-1,I) + Y(L+1,I) + Y(L,I+1) + Y(L,I+1) - 4 * Y(L,I)_{old} \right\}$$

The value of $\alpha$ and the number of times of application were determined by trial and error.

The "smoothed" boundary points could not be determined from the five point diffusion operator since one of the required smoothing points would be outside the mesh region. Their values were determined from the intersection of the lines defined by the "smoothed" interior mesh points and the boundaries.

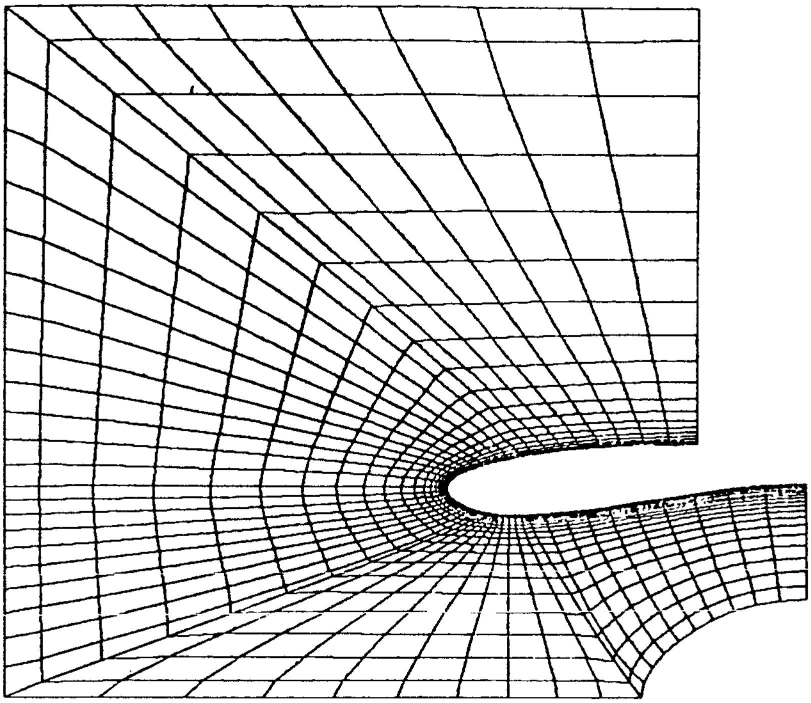


$$X(L,I)_{new} = \alpha \left\{ X(L-1,I) + X(L+1,I) + X(L,I-1) + X(L,I+1) - 4 * X(L,I)_{old} \right\}$$

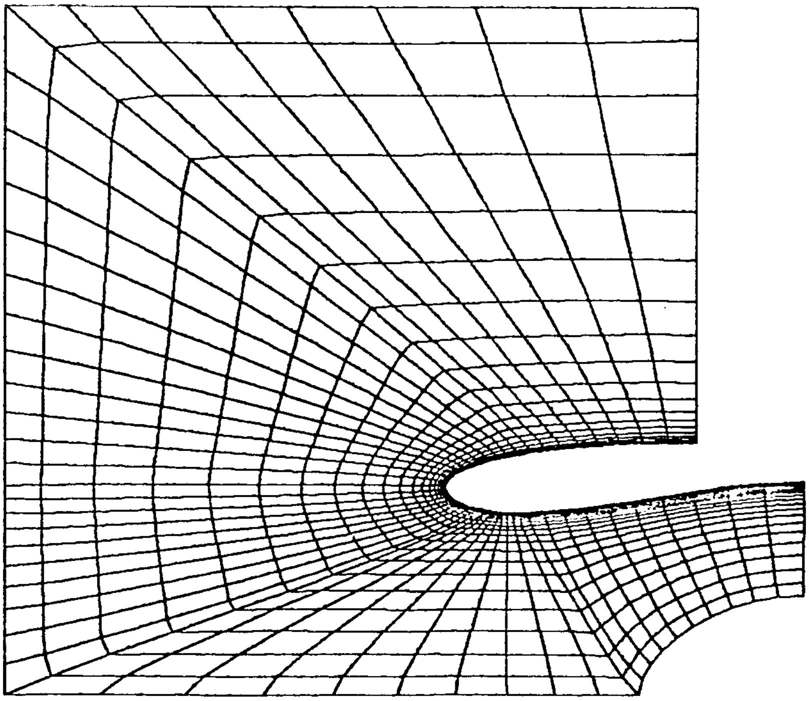
$$Y(L,I)_{new} = \alpha \left\{ Y(L-1,I) + Y(L+1,I) + Y(L,I+1) + Y(L,I+1) - 4 * Y(L,I)_{old} \right\}$$

FIVE POINT DIFFUSION OPERATOR

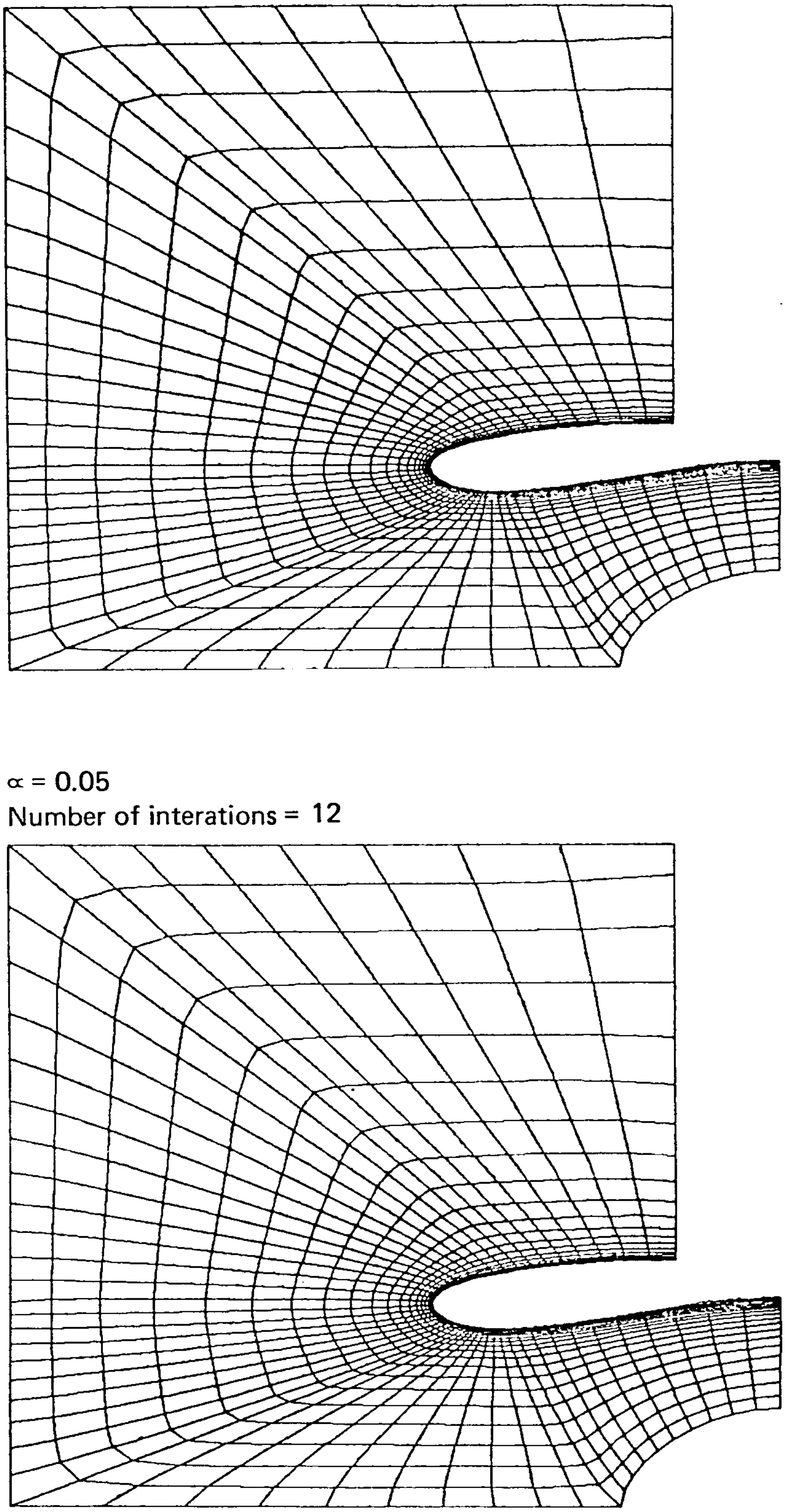
$\alpha = 0.05$
Number of interations = 1



$\alpha = 0.05$
Number of interations = 3



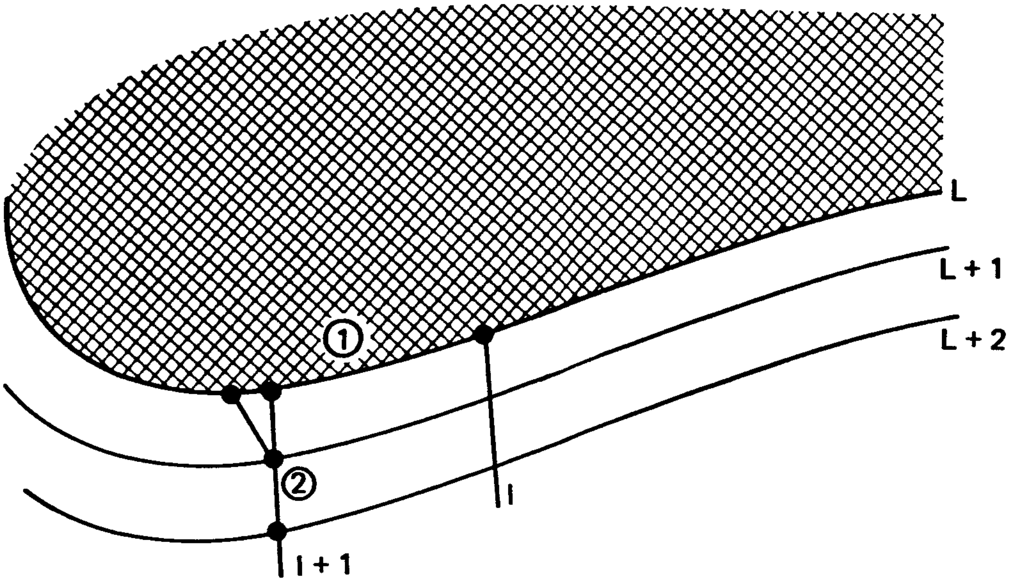
"SMOOTHED" COMPUTATIONAL MESH



$\alpha = 0.05$
Number of interations = 6

$\alpha = 0.05$
Number of interations = 12

"SMOOTHED" COMPUTATIONAL MESH



slope of segment ①

$$M_1 = \frac{Y(L, I) - Y(L, I+1)}{X(L, I) - X(L, I+1)}$$

equation of segment ①

$$Y - Y(L,I) = M_1\left\{X - X(L,I)\right\} \qquad \textcircled{2}$$

slope segment ②

$$M_2 = \frac{Y(L+1, I+1) - Y(L+2, I+1)}{X(L+1, I+1) - X(L+2, I+1)}$$

equation of segment ②

$$Y - Y(L+1, I+1) = M_2\left\{X - X(L+1, I+1)\right\} \qquad \textcircled{1}$$

since a line thru segment ② intersects segment ①, the X's and Y's of equations ① & ② equal each other.

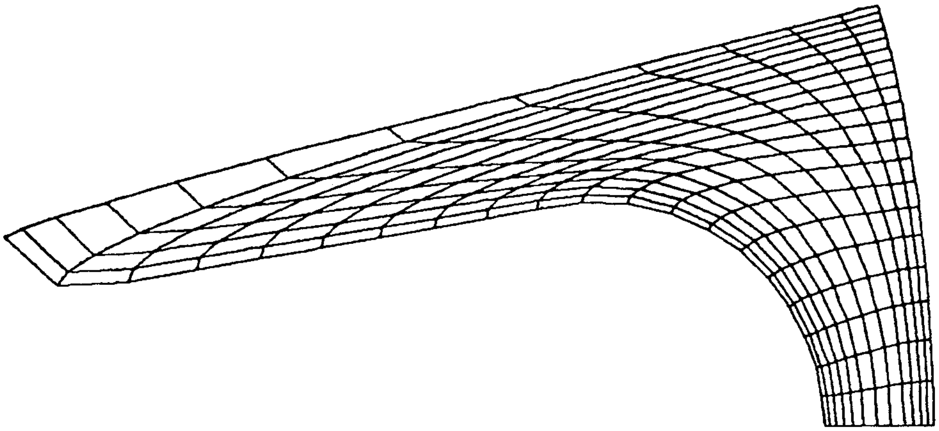
Solving for X:

$$X(L, I+1)_{new} = \frac{M_1\left\{X(L, I)\right\} - M_2\left\{X(L+1, I+1)\right\} + Y(L+1, I+1) - Y(L, I)}{M_1 - M_2}$$

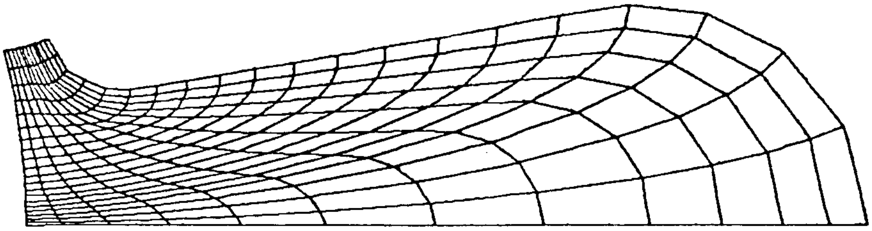
Solving for Y:

$$Y(L, I+1)_{new} = M_1\left\{X(L, I+1)_{new} - X(L, I)\right\} + Y(L, I)$$

Example meshes for the last radial mixer lobe cross section and one subsonic inlet cross section.

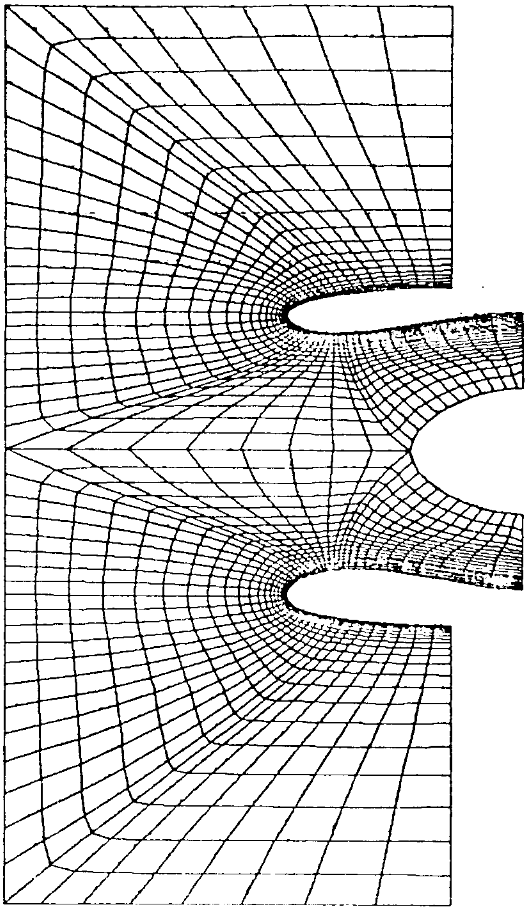


SECONDARY LOBE - PLANE 120
GEOMETRIC PROGRESSION   LOWER BOUNDARY  0.9500  UPPER BOUNDARY  1.1000
SLOPE SCALER  LOWER BOUNDARY  1.0500  UPPER BOUNDARY  1.1000



PRIMARY LOBE - PLANE 120
GEOMETRIC PROGRESSION   LOWER BOUNDARY  1.1000  UPPER BOUNDARY  1.1000
SLOPE SCALER  LOWER BOUNDARY  1.1000  UPPER BOUNDARY  1.1000

*Example Mesh for Last Mixer Plane.*



EXAMPLE MESH FOR THE SUBSONIC INLET

## Conclusions

The method of Smith and Wiegel can be used to generate meshes for mixer lobes and subsonic inlets that are compatible with flow analysis codes requiring a boundary fitted coordinate system. Successful application of this mesh generator required development of procedures to distribute the mesh points along the boundaries, to regulate the dependence of the connecting function to the local boundary slope, to concentrate the mesh into regions of special interest, and to modify the mesh grid so that it possessed a smooth progression of cell metrics and cell volumes in all directions. The method of Smith and Wiegel when coupled with the extensions mentioned above has proven to be easy to use and control for the inlet and mixer lobe geometries investigated.

The next step is the formulation of a truncation error monitor for arbitrary meshes. This monitor will define where in an analysis domain the grid length scales must be changed and by what amount in order to equalize truncation errors over the entire analysis domain. Once these errors have been equalized, this same monitor will use several levels of grid distribution (of the above analysis grid) to then make estimates of the absolute truncation error spectrum. This work is currently under contract with the NASA Langley Research Center.

# A VARIABLE COEFFICIENT APPROACH TO BOUNDARY-FITTED COORDINATES

by
Thomas T. Bowman
University of Florida

Prepared under U.S. Navy Contract 1800-0832-79 issued by
David W. Taylor Naval Ship Research and Development Center

The author has investigated the use of the generating equations

$$(\sigma\xi_x)_x+(\sigma\xi_y)_y=0$$

$$(\tau\eta_x)_x+(\tau\eta_y)_y=0$$

where $\sigma$ and $\tau$ are strictly positive functions. Much of the work in boundary-fitted coordinates can be presented naturally in a unified approach using variable coefficients. The advantages of this approach are:

(1) The generating equations have both a maximum and minimum property.

(2) The equations have the physical interpretation as the heat equation with variable thermal conductivity constants which leads to an intuitive concept of controlling the $\xi$ and $\eta$ lines.

The method has been implemented on a CDC 6700 computer at DTNSRDC. The adoption of the variable coefficient approach can be accomplished by limited changes in most programs which generate boundary-fitted coordinates.

The author also has some new results on generating orthogonal meshes with prescribed boundary values.

# GRID AND METRIC GENERATION ON THE ASSEMBLY OF LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATIONS[†]

A. J. BAKER & P. D. MANHARDT

UNIVERSITY OF TENNESSEE/KNOXVILLE,
AND COMCO, INC, KNOXVILLE, TN

## ABSTRACT

The generation of metric coefficients of the coordinate transformation from a generally curved-sided domain boundary to the unit square (cube) is required for efficient solution algorithms in computational fluid mechanics. An algebraic procedure is presented for establishment of these data on the union of arbitrarily selected sub-domains of the global solution domain. A uniformly smooth progression of grid refinement is readily generated, including multiple specification of refined grids for a given macro-element domain discretization. The procedure is illustrated as generally applicable to non-simply connected domains in two- and three-dimensions.

[†] Research principally supported by USAF Grant No. AFOSR-79-0005.

# COMPUTATIONAL REQUIREMENT

## NAVIER-STOKES EQUATIONS

$$L(q_i) = \frac{\partial q_i}{\partial t} + \frac{\partial}{\partial x_j}\left[u_j q_i + f_{ij}\right] = 0$$

$$\ell(q_i) = a_1 q_i + a_2 \frac{\partial q_i}{\partial x_j}\hat{n}_j + a_3 = 0$$

## COORDINATE TRANSFORMATION

$$x_i = x_i(\eta_j) \qquad \frac{\partial}{\partial x_j} = \frac{\partial}{\partial \eta_k}\left[\quad\right]\frac{\partial \eta_k}{\partial x_j}$$

$$J^{-1} = \left[\frac{\partial \eta_k}{\partial x_j}\right] \qquad \bar{u}_k = \frac{\partial \eta_k}{\partial x_j} u_j$$

## NUMERICAL SOLUTION ALGORITHM

$$S_e\Big[\{DET\underline{J}\}_e^T[M3000]\{QI\}_e^{\blacktriangledown} - \{UBAR\underline{K}\}_e^T[M30\underline{K}0]\{QI\}_e$$

$$-\{ETA\underline{KL}\}_e^T[M30\underline{K}0]\{F\underline{L}I\}_e\Big] \equiv \{0\}$$

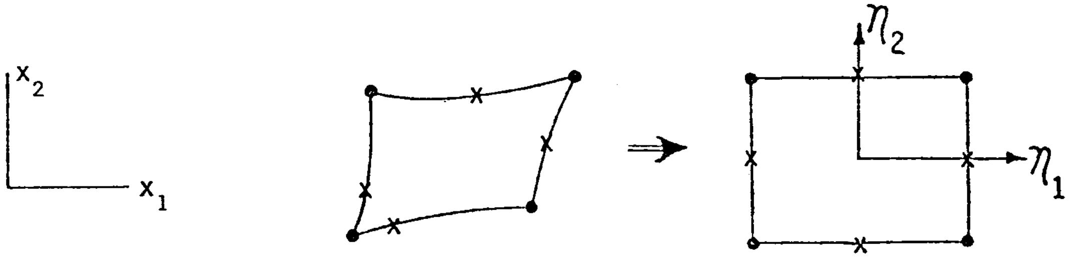
## DISCUSSION

The Navier-Stokes equations contain the vector divergence operator. The required transformation projects $x_i$ onto $\eta_j$ with coordinate surfaces defined coincident with solution domain boundaries. The Cartesian description of dependent variables is retained, while the convection velocity is expressed in contravariant scalar components. The numerical solution implementation requires nodal distributions of components of the forward and inverse Jacobins, and $\underline{J}$, $\underline{K}$, and $\underline{L}$ are tensor summation indices.

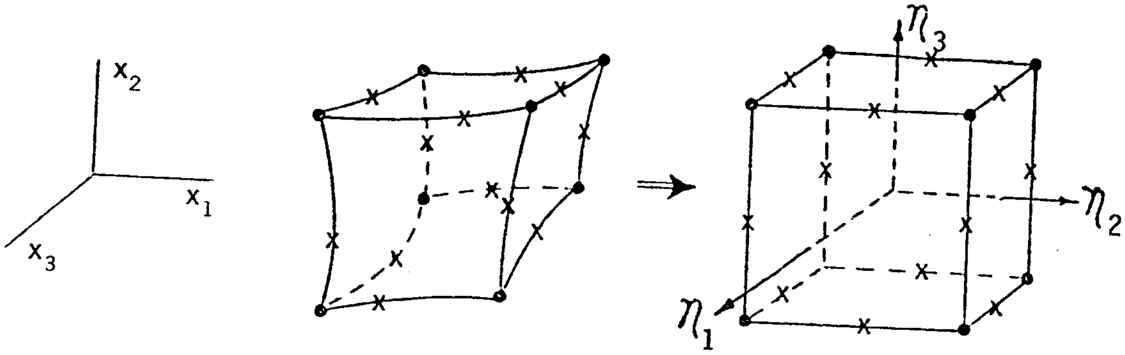
# LOCALLY BI-QUADRATIC COORDINATE TRANSFORMATION

PHYSICAL DOMAIN TRANSFORMED DOMAIN

$$x_i \equiv \{N_2(\vec{\eta})\}^T\{XI\}_e$$



Two-Dimensional

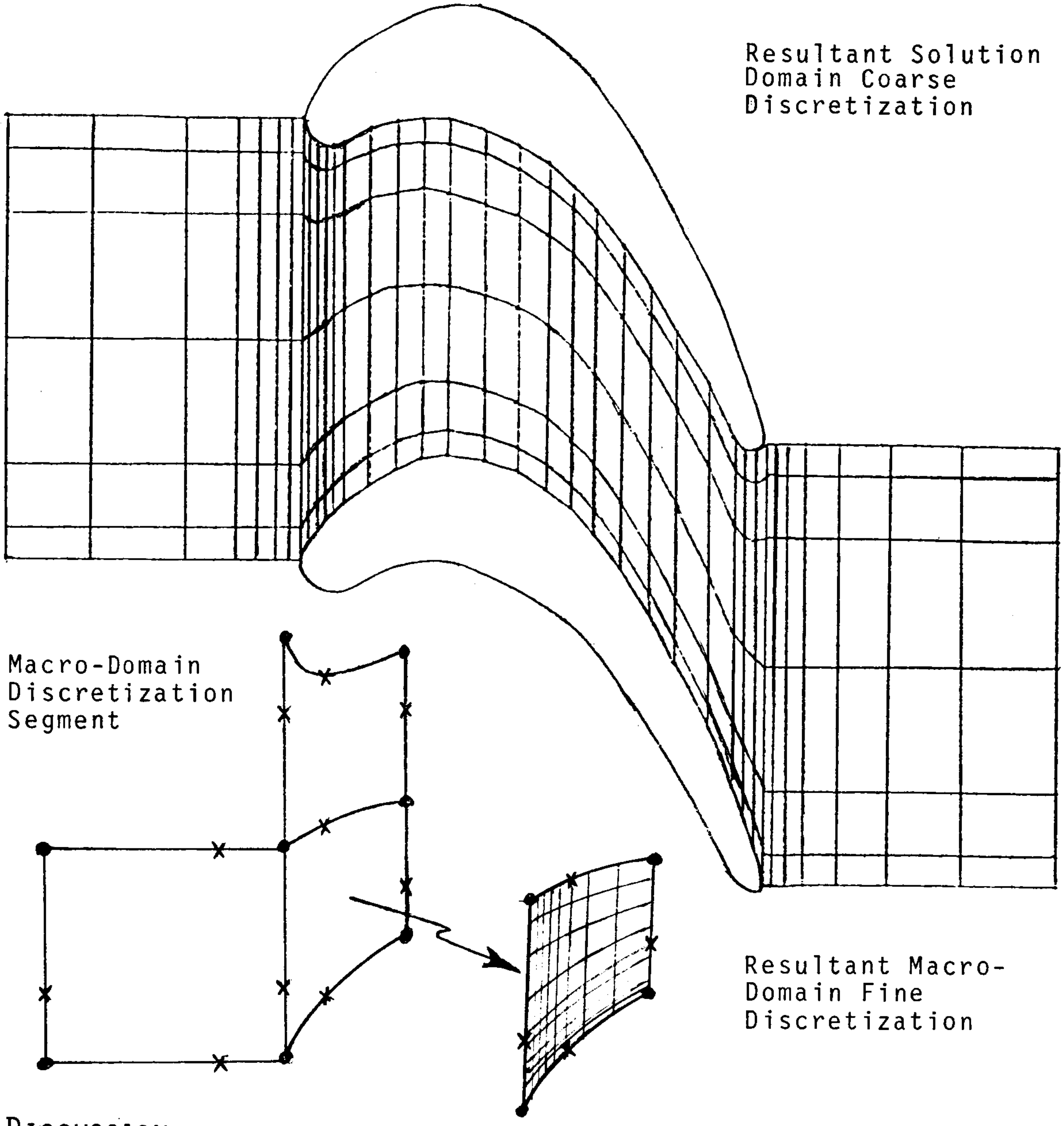


Three-Dimensional

## DISCUSSION

The bi-quadratic cardinal basis $\{N_2(\vec{\eta})\}$ transforms the vertex and non-vertex node coordinate description of a smooth region of $R^n$ onto the unit square or cube spanned by the locally rectangular Cartesian coordinate system $\vec{\eta}$. The inverse transformation $J^{-1}$ is non-singular for a range of non-midpoint definitions of the non-vertex node coordinates (x), yielding a non-uniform discretization on $R^n$.

# EXAMPLE: COMPRESSOR BLADE ROW


Resultant Solution Domain Coarse Discretization
Macro-Domain Discretization Segment
Resultant Macro-Domain Fine Discretization

## DISCUSSION

Three of the ten macro-domains, used to form the blade row discretization, are shown. The non-midside location of non-vertex nodes produces the non-uniform grid, only a few gridlines of which are shown. The inset illustrates a fine fiscretization of one macro-domain. The coordinates of all nodes on boundaries of macro-domains are unique.

# DETAILS OF THE COORDINATE TRANSFORMATION

NODAL COORDINATES $\{XI\}$:

$$x_i \equiv \{N_2(\eta_j)\}^T \{XI\}_e$$

WHERE:

$$\{N_2(\eta_i)\} \equiv \frac{1}{4} \left\{ \begin{array}{l} (1 - \eta_1)(1 - \eta_2)(-\eta_1 - \eta_2 - 1) \\ (1 + \eta_1)(1 - \eta_2)(\ \eta_1 - \eta_2 - 1) \\ (1 + \eta_1)(1 + \eta_2)(\ \eta_1 + \eta_2 - 1) \\ (1 - \eta_1)(1 + \eta_2)(-\eta_1 + \eta_2 - 1) \\ 2(1 - \eta_1^2)(1 - \eta_2) \\ 2(1 + \eta_1)(1 - \eta_2^2) \\ 2(1 - \eta_1^2)(1 + \eta_2) \\ 2(1 - \eta_1)(1 + \eta_2^2) \end{array} \right\}$$

JACOBIANS

$$J \equiv \left[\frac{\partial x_i}{\partial \eta_j}\right] = J(\eta_j,\ XI)$$

$$J^{-1} \equiv \left[\frac{\partial \eta_j}{\partial x_i}\right] = \frac{1}{\det J}\ [\text{cofactors of } J]$$

$$= J^{-1}(\eta_j,\ XI)$$

## DISCUSSION

Within a macro-domain, the components of both J and $J^{-1}$ are continuous functions of $\eta_j$ and the global macro-node coordinate pairs (triples) $\{XI\}$, $1 \leq I \leq n$. Each global coordinate $x_i$ possesses an independent transformation; the corresponding Jacobian must be of rank n to assure existence of $J^{-1}$. Once the matrix elements of $\{XI\}$ are defined, selection of any coordinate $(\eta_1, \eta_2)$ defines a unique coordinate pair $(x_1, x_2)$, i.e., a mesh point on the refined grid in physical space.

# GENERATIONS OF ORTHOGONAL SURFACE COORDINATES*

F. G. Blottner and J. B. Moreno
Sandia National Laboratories†
Albuquerque, NM 87185

An orthogonal surface-oriented coordinate system has been developed for three-dimensional flows where the computational domain normal to the surface is small. With this restriction the coordinate system requires orthogonality only at the body surface. The coordinate system is as follows: one coordinate measures distance normal to the surface while the other two coordinates are defined by an orthogonal mesh on the surface. One coordinate is formed by the intersection of the body surface and the meridional planes as illustrated in Figure 1 and gives the $\theta$ = constant lines. The other coordinate $\xi$, which is nondimensionalized with a characteristic length of the body geometry, measures the distance along the body surface when $\theta = 0$. This coordinate system has been utilized in boundary layer flows[1,2] and for the hypersonic viscous shock-layer problem.[3]

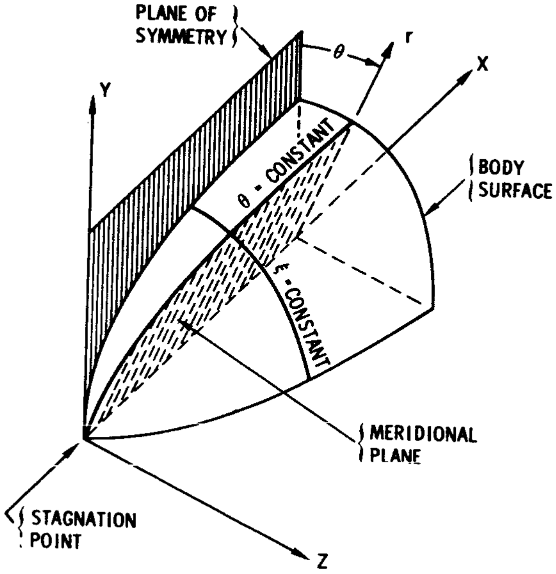


Figure 1. Surface Coordinate System.

Two methods have been developed for generating the surface coordinates. The first method uses the orthogonal condition in finite-difference form to determine the surface coordinates with the metric coefficients and curvature of the coordinate lines calculated numerically. The second method obtains analytical expressions for the metric coefficients and for the curvature of the coordinate lines. Both methods assume the body surface is defined in terms of a cylindrical coordinate system where $r = r(x,\theta)$. The surface inclinations $\phi_1$ and $\phi_2$ as illustrated in Figure 2 are determined from

$$\tan \phi_1 = \left(\frac{\partial r}{\partial x}\right)_\theta \quad \text{and} \quad r \tan \phi_2 = -\left(\frac{\partial r}{\partial \theta}\right)_x$$

and are known quantities.

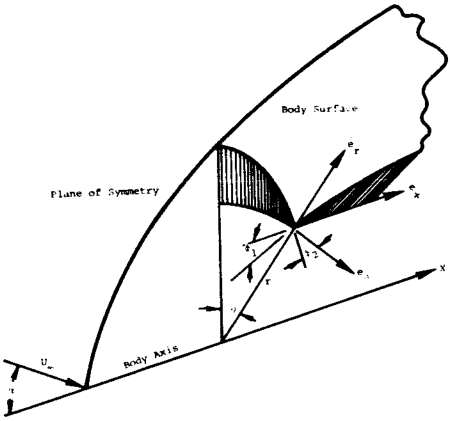


Figure 2. Angles $\phi_1$ and $\phi_2$ Defined in the Cylindrical Coordinate System.

* This work was supported by the U. S. Department of Energy under contract DE-AC04-76-DP00789.

† A U. S. Department of Energy Facility.

In the numerical method,[1,2] the orthogonal condition for the surface coordinates results in the relation

$$dx = \lambda \, d\theta \qquad \text{(along } \xi = \text{constant)}$$

where

$$\lambda = r \tan \phi_1 \tan \phi_2/(1 + \tan^2 \phi_1)$$

The equation of the surface provides the relation

$$dr = \tan \phi_1 \, dx - r \tan \phi_2 \, d\theta$$

The surface coordinate $\xi$ is determined numerically from the foregoing equations by assuming a value of $d\theta$ and marching away from $\theta = 0$ to determine the values of x and r. In addition the metric coefficients are determined numerically from

$$h_\xi = ds/d\xi$$

$$h_\omega = dt/d\omega$$

where

$$\omega = \theta/2\pi$$

$$ds^2 = dx^2 + dr^2$$

$$dt^2 = ds^2 + r^2 d\theta^2$$

The curvature of the coordinate lines are determined from

$$K_\xi = \frac{1}{h_\xi h_\omega} \frac{\partial h_\xi}{\partial \omega} \text{ for } \omega = \text{constant}$$

$$K_\omega = \frac{1}{h_\xi h_\omega} \frac{\partial h_\omega}{\partial \xi} \text{ for } \xi = \text{constant}$$

with the derivatives replaced with midpoint difference relations.

•In the second method[3], an analytical expression is developed for $h_\omega$ as follows:

$$h_\omega = 2\pi r \, (1 + \cos^2\phi_1 \tan^2\phi_2)^{1/2}$$

A differential equation results for the other metric coefficient as follows:

$$\frac{1}{h_\xi} \frac{dh_\xi}{d\omega} = 2\pi r \cos^2\phi_1 \tan \phi_2 \left(\frac{\partial \phi_1}{\partial x}\right)_\theta$$

This equation is integrated along $\xi$ = constant lines on the surface from the initial condition $h_\xi$ = 1 at $\theta = 0$. The substitution of foregoing equations into the equations for $K_\xi$ and $K_\omega$ give analytical expressions for the curvature of the coordinate lines. In evaluating these relations, the variations of x and $\theta$ along the $\xi$ = constant coordinate must be known.

A sphere at angle of attack as shown in Figure 3 is used to illustrate the computation of the surface coordinates with both methods. The surface coordinates on the sphere as viewed from the side are illustrated in Figure 4. The $\xi$ = constant lines result from planes intersecting the sphere with these planes passing through the line which is normal to the plane of symmetry and is located at

$$x/a = \sqrt{1 - (b/a)^2}$$

$$y/a = (x/a)^2/(b/a)$$

The metric coefficients for this coordinate system are given in Figures 5 and 6 with good agreement between the two methods. The curvature of the coordinate lines are given in Figures 7 and 8. It is noteworthy that $K_\xi$ is independent of $\xi$. The differences evidenced in Figure 8 can be partially attributed to the numerical evaluation of $K_\omega$ being at one-half mesh space locations away from the $\xi$ indicated.

The numerical method of generating the orthogonal surface coordinates has been applied to ellipsoids, paraboloids and elliptic-paraboloids. The coordinates on an ellipsoid are illustrated in Figure 9. The second method or analytical approach has only been developed for the sphere.

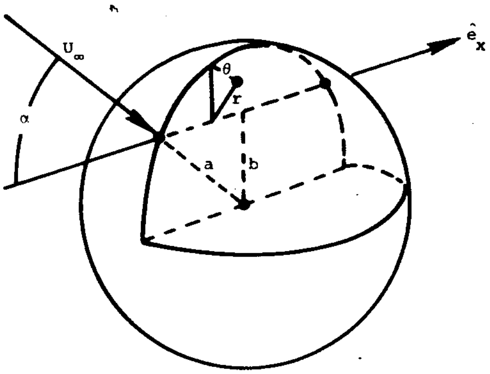


Figure 3. Cylindrical Coordinate System for "Sphere at Angle of Attack".

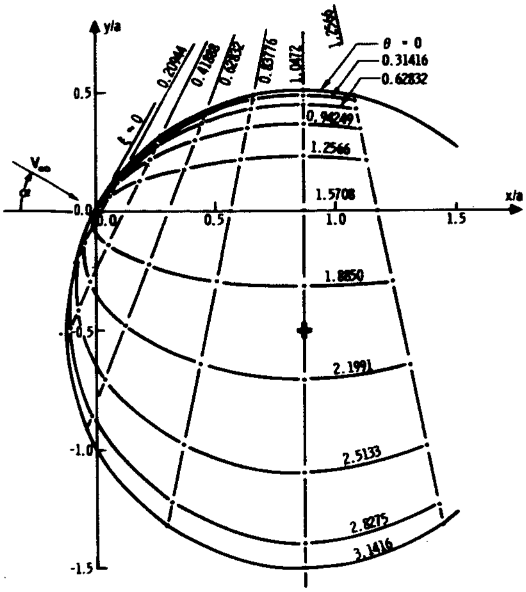


Figure 4. Coordinates on a Sphere (α = 30°)

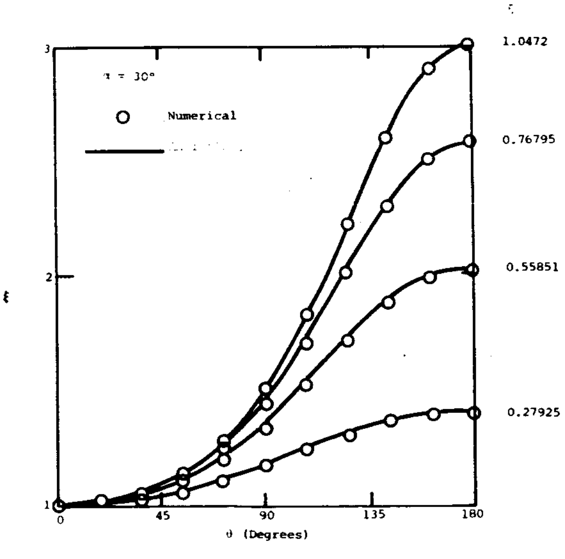


Figure 5. Variation of Metric Coefficient $h_\xi$.

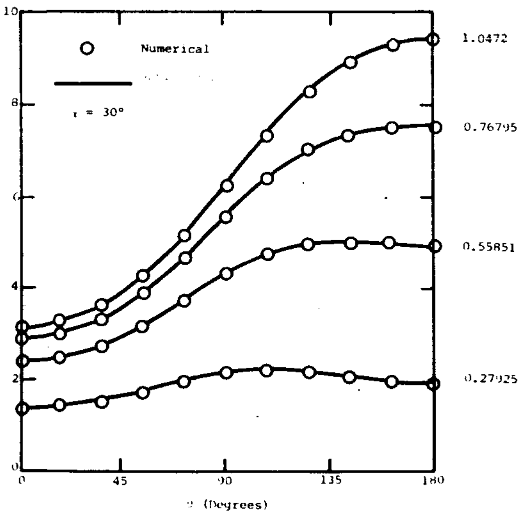


Figure 6. Variation of Metric Coefficient $h_\omega$.

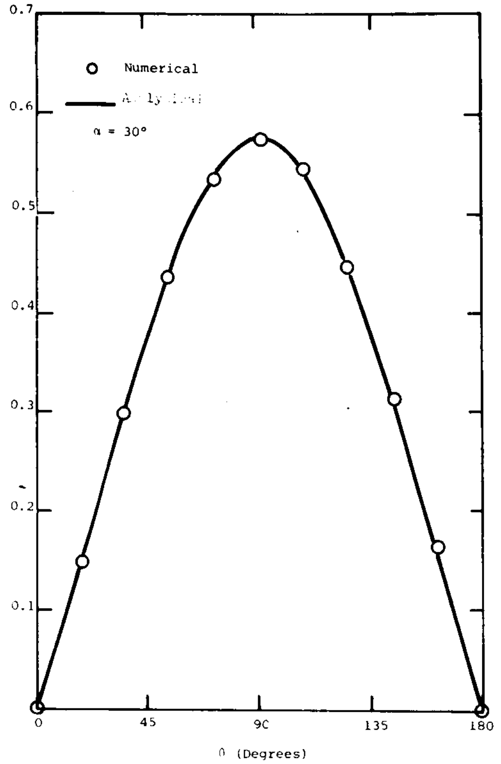


Figure 7. Geodesic Curvature of Lines of Constant θ.

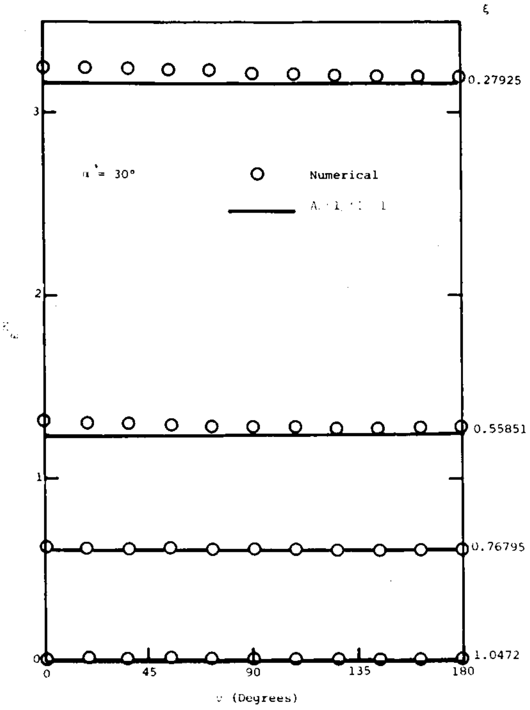


Figure 8. Geodesic Curvature of Lines of Constant ξ.

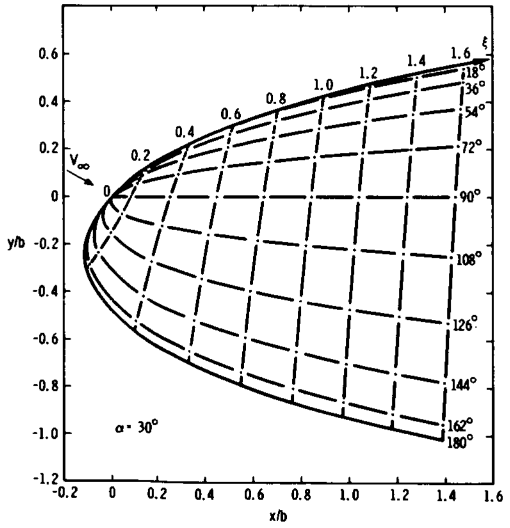


Figure 9. Surface Coordinates on Ellipsoid (b/a = 1/4)

References:

1. F. G. Blottner and Molly A. Ellis, "Finite-Difference Solution of the Incompressible Three-Dimensional Boundary Layer Equations for a Blunt Body," Computers and Fluids, Vol. 1, pp 133-158 (1973).

2. F. G. Blottner and Molly A. Ellis, "Three-Dimensional, Incompressible Boundary Layer on Blunt Bodies. Part I: Analysis and Results," Sandia Laboratories Research Report SLA-73-0366, April 1973.

3. J. B. Moreno, "Formulation of the Three-Dimensional Hypersonic Viscous Shock-Layer Problem Part I: Governing Equations and Geometrical Formulas," Sandia Laboratories Research Report SLA-73-0030.

# CONSTRUCTION OF COMPOSITE THREE-DIMENSIONAL GRIDS FROM SUBREGION GRIDS GENERATED BY ELLIPTIC SYSTEM

by

P. D. Thomas

Paper not available for printing.

# SOME ASPECTS OF ADAPTING COMPUTATIONAL MESH TO COMPLEX FLOW DOMAINS AND STRUCTURES WITH APPLICATION TO BLOWN SHOCK LAYER AND BASE FLOW

by

C.K. Lombard, M.P. Lombard, G.P. Menees, and J.Y. Yang

The proposed paper treats several practical aspects connected with the notion of computation with flow oriented mesh systems. Simple, effective approaches to the ideas discussed are demonstrated in current applications to blown forebody shock layer flow and full bluff body shock layer flow including the massively separated wake region.

The first task in constructing an adaptive mesh is to identify the gross flow structures that are to be captured on the mesh and to work out a grid topology that conforms to them. Among the properties the mesh topology ought to admit are both computational accuracy and algorithmic compatibility. Both these properties are served by grids that feature large connected segments of natural or computational boundaries fitted by mesh surfaces or curves of constant coordinate. But it is neither necessary or always desireable that the entire surface of a particular boundary feature be fitted by a single surface segment of one family of coordinates. For accuracy, convenience, and particularly from the point of view of modern algorithms that embody such features as vector organization, spatial splitting, and implicit solution, it is very desireable that the mesh be composed of identifiable continuous grid lines, not necessarily of homogeneous coordinate type, that run from boundary to boundary.

These notions are illustrated in the application to high Reynolds number full bluff body flow in axisymmetry. Here the basic structure of the turbulent flow is well known, Figure 1. The computational mesh that we have adapted to the flow is shown in Figure 2.

We note that in the mesh shown the computational boundaries — axis of symmetry, bowshock, body, and outflow plane are all fitted by continuous grid lines. The mesh is so constructed as to be flow aligned over the four principal regions — forebody shocklayer, base recirculation, outer inviscid wake, and inner turbulent viscous wake. We note the wrap around mesh provides continuity of the boundary layer and shear layer in the aft expansion zone. The continuity of the mesh coordinate topology is broken in the recompression zone which embeds a saddle surface of the turbulent flow solution at the interface of the recirculant base flow and downstream viscous wake. The singular topology of the mesh in the base recompression zone is illustrated in Figure 3. The viscous wake core box of the mesh, which provides continuity across the viscous-inviscid wake shear layer, can be regarded as a separate sheet of the topology with a cut taken along a line from the singular point down through the recompression zone to the wake axis.

The cut forms part of a set of construction lines embedded in the mesh, Figure 4. It is central to the method described that these lines which largely define the base mesh structure are also representative of the flow structures which the mesh is to fit. Thus in the approach presented here the construction lines serve the role of supplemental imaginary boundaries along which mesh nodes are distributed according to the usual criteria on ordinary boundaries. The resulting bounded domains can then be filled in with computational grid by any of a large variety of means, for example[1,2,3,4].

The particular grid shown in Figure 2 is quite adequate in concept, though not optimized in detail, and was simple constructed in a single pass using one dimensional distributions along straight coordinate lines between boundary points. Where non-uniform distributions have been required they have been conveniently accomplished using a universal stretching function due to Vinokur[5]. In the program, for the stretching function as we have adapted and use it, the total interval along the coordinate line and the (approximate) first mesh spacings from either end of the interval are specified. The function then returns the distribution between boundary points. As convenient, the stretchings are done variously in X, Y, or S (arc length). The actual X and Y coordinates of mesh points are then found by the functional relationships of points on the given coordinate curve, which of course can be piecewise defined. Where fictitious boundary lines are to be embedded in the mesh actual boundary points are defined on the connecting coordinate lines at half first-mesh-cell intervals away from the fictious lines.

A virtue of meshes constructed of distributions along analytically defined coordinate curves, and particularly straight lines, is that differential displacements of boundary points are readily functionally transformed through kinematic relations into corresponding displacements of the intervening grid points so as to leave invariant the relative distributions of mesh points along the given coordinate curves. For the mesh shown in Figure 2, we presently use this property to analytically deform the outer flow portion of the mesh in relative conformity with the moving, fitted bowshock.

In a similar manner it is intended in future work to differentially adapt the interior base mesh to the changing flow solution by moving the underlying construction lines. A central requirement to do this is to define relationships tying the construction lines to the base flow solution. In this regard it is intended that the X coordinate of the mesh singularity coorespond to the axial location of maximum wake pressure. Presumably, the Y coordinate of the singularity which lies on the construction line through the viscous-inviscid wake shear layer ought to be determined from a fit to the axial velocity gradient.

Along the same lines, however, we have developed an adaptive mesh for the blown forebody shock layer which is intended to represent flow over an ablating body. Here we wish to distribute points in predetermined ways in the blown layer, the shear layer interface, and in the outer flow region. In this case a construction line demarking the interface between the blown and outer flow regions can readily and unambiguously be fitted to the zero of the stream function based on mass flux and this is what we have done.

We note in connection with the blown shock layer that the associated flow has regions of steep gradient in density, velocity, mass flux, and temperature and that these properties by no means vary together. We take it that an accurate calculation ought to resolve all these features. Thus we think for this application a mesh distribution approach based on the integral of gradient of a single flow property such as Dwyer[6] has demonstrated is not evidently optimum. A similar distribution based on weighted gradients is certainly feasible but this would appear to be more tedious to implement than a compromise ad hoc distribution tied to key features of the flow structure as we have done. In the paper we shall present curves showing the variation of relevant flow properties across a blown shock layer and show how the simple ad hoc distribution approach we use results in satisfactory resolution of all properties.

## References

1. Smith, R.E. and Weigle, B.L., "Analytic and Approximate Boundary Fitted Coordinate Systems for Fluid Flow Simulation," AIAA-80-0192, AIAA 18th Aerospace Sciences Meeting, Pasadena, CA, Jan. 14-16, 1980.

2. Eiseman, Peter R., "Coordinate Generation with Precise Controls," Seventh International Conference on Numerical Methods in Fluid Dynamics, Stanford University and NASA-Ames Research Center, June 23-27, 1980.

3. Thompson, J.F., Thames, F.C., and Mastin, C.W., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate Systems for Fields Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, 15, 299, 1974.

4. Middlecoff, J.F. and Thomas, P.D., "Direct Control of the Grid Point Distribution in Meshes Generated by Elliptic Equations," AIAA-79-1462, AIAA Computational Fluid Dynamics Conference, Williamsburg, VA, July 23-25, 1979.

5. Vinokur, Marcel, "On One-Dimensional Stretching Functions for Finite-Difference Calculations," Final Technical Report for Period July 1, 1978 to June 30, 1979, Grant No. NSG 2086, The University of Santa Clara, CA.

6. Dwyer, H.A., Kee, R.J., and Sanders, B.R., "An Adaptive Grid Method for Problems in Fluid Mechanics and Heat Transfer," AIAA-79-1464, AIAA Computational Fluid Dynamics Conference, Williamsburg, VA, July 23-25, 1979.

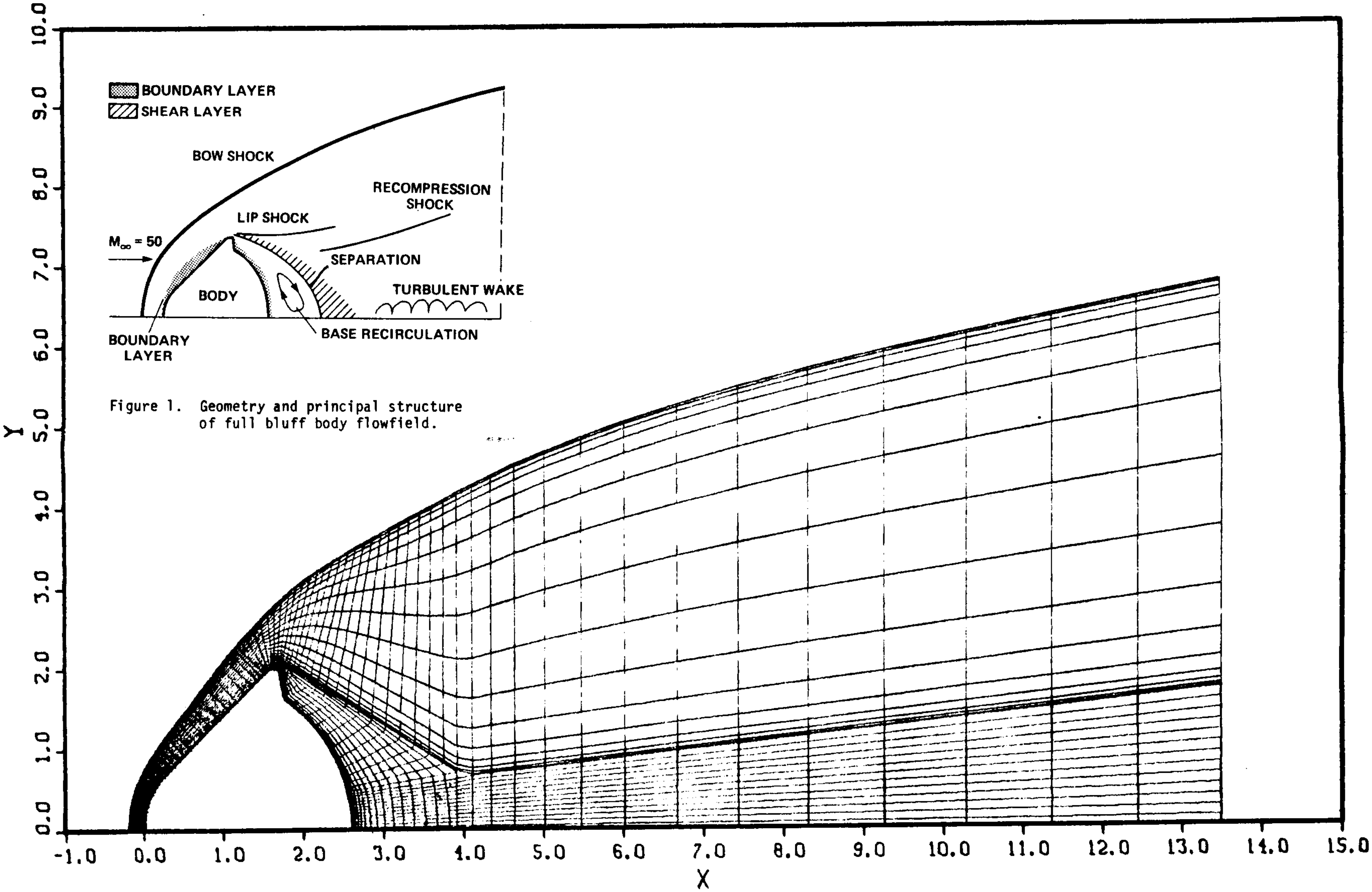


Figure 1. Geometry and principal structure of full bluff body flowfield.

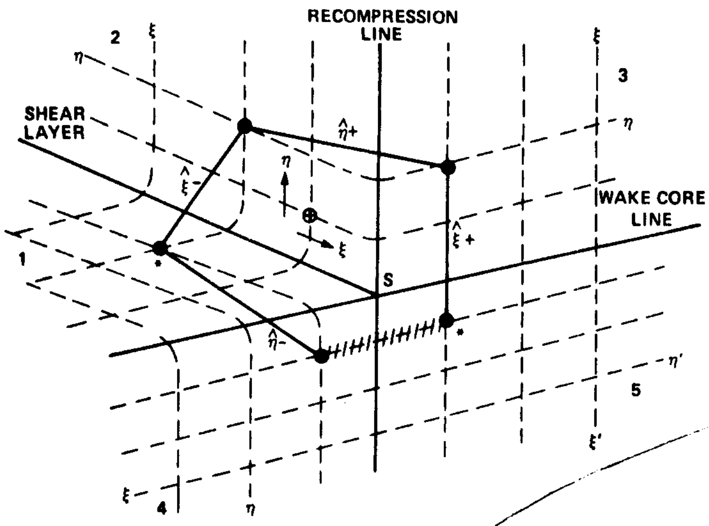


Figure 3. Base region mesh detail showing topology of the coordinate lines in the vicinity of the singular point s .

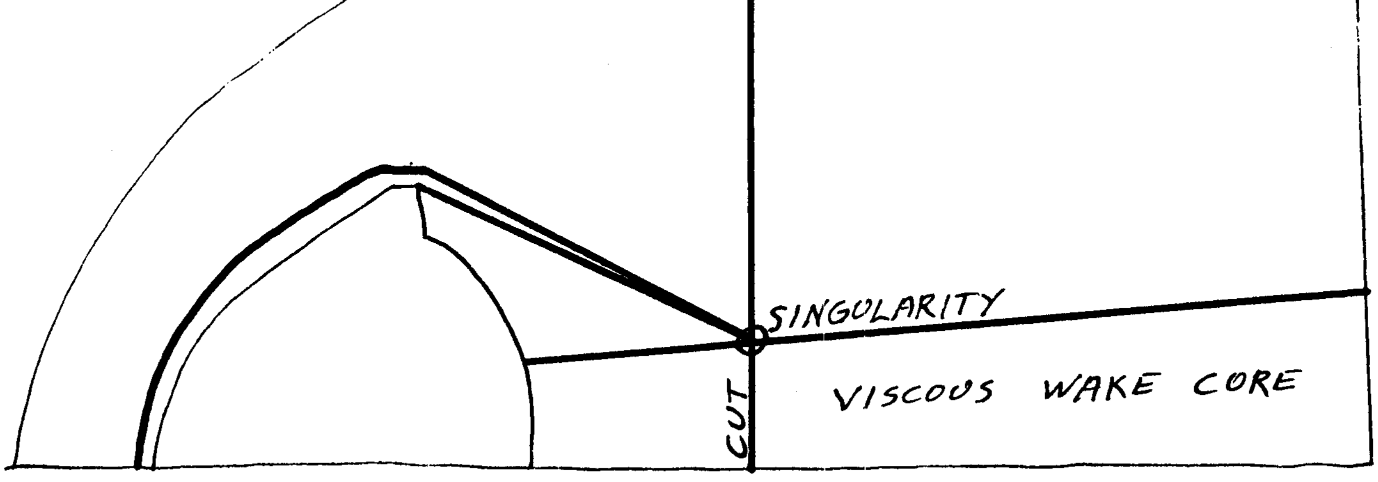


FIG.4 MESH CONSTRUCTION LINES

# GRID GENERATION USING COARSE, SMOOTH FINITE ELEMENTS

by Lawrence J. Dickson*
Mail Stop FS-10
Dept. of Aeronautics & Astronautics
University of Washington
Seattle, WA 98195

## SUMMARY

### I. Approach

The grid generation problem lends itself to the use of finite elements and variation equations.

(1) Grids are usually generated as smooth solutions to "nice," elliptic differential equations--just the equations well suited to variational methods.

(2) The use of smooth finite elements gives the grid a functional expression, which can be examined, evaluated, manipulated, and modified naturally and cheaply.

(3) The "grid equations" are chosen for their qualitative character. Exactitude of solutions does not matter as long as this is preserved. As a result, extremely coarse (cheap) finite elements may generate a grid of high quality, if the boundary conditions are well parameterized.

### II. Results

I succeeded in demonstrating the following.

(1) Grid-quality solutions of a wide variety of equations--(direct) Laplace's, biharmonic, Helmholtz, even nonlinear--can be generated to fit reasonable functional boundary conditions in 2D using very coarse rectangular finite elements, often 6x3 $C^2$ bicubic. I even tried some "wavy" operators (with no natural variational expression) to demonstrate the method's versatility. I did not try the inverse Laplace equation, but I expect no problem

*This work was supported by the Boeing Commercial Airplane Co.

but cost.

(2) The finite element grids can be refined, locally modified and "fine-tuned" using a simple, cheap composition-of-functions approach, without having to solve the differential equation repeatedly.

# Generation of Boundary and Boundary-Layer Fitting Grids

by

C. M. Ablow and S. Schechter
SRI International

## ABSTRACT

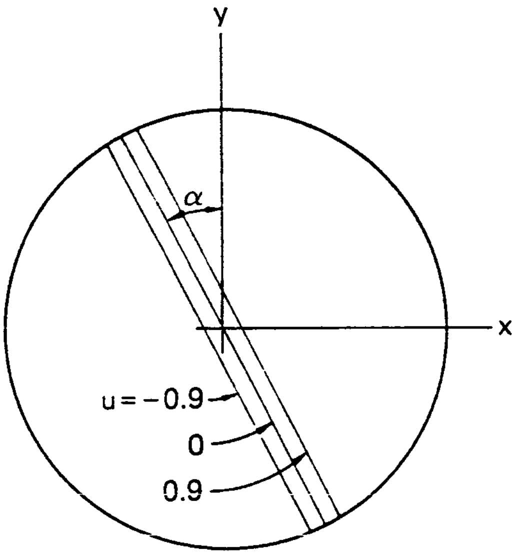
A grid that improves the accuracy and speed of computation with a given finite difference approximation to a boundary value problem for a differential equation is more satisfactory than other grids. A best method of grid generation will therefore depend on the problem domain, the solution, and the difference scheme.

An automatic generator for the grid that minimizes the truncation error of a given difference scheme for two-point boundary value problems over a finite one-dimensional interval has been previously presented.* This truncation error minimizing (TEM) generator changes the independent variable to one in which uniformly spaced nodes fit the boundaries and cluster in any boundary layers where the solution has a sharp variation. The number of nodes and the complexity of the calculation are known in advance so that the time and cost of the calculation can be estimated. Other generators producing grids that equally distribute measures of the solution curve arc length or length and curvature were found to be about as accurate as the TEM generator but more easily implemented. The arc length coordinate can also be defined as the transformation that minimizes the sum of the squares of the derivatives of the dependent and independent variables, a definition that readily generalizes to higher dimensions.

Experience with two-dimensional grid generation, as applied to a Dirichlet problem for the Poisson equation on the unit disc, is presented. The example has an analytic solution with sharp variation across a

*C. M. Ablow, S. Schechter, and W. H. Zwisler, "Node Selection for Two-Boundary Value Problems," sumbitted to Computational Physics.

diameter of the disc. The grid is uniformly rectangular on the unit square in the transformed coordinates. Transformations were chosen to minimize the sum of the squares of the derivatives of the dependent variable and of the dependent and originally independent variables. The TEM transformation was judged too complex to be practical. The results show that the grid fits the boundaries, clusters about the boundary layer, and rotates into alignment with it as desired.



$$u_{xx} + u_{yy} = 2p^2 \operatorname{sech}^2 pz \tanh pz$$

$$u = \tanh pz$$

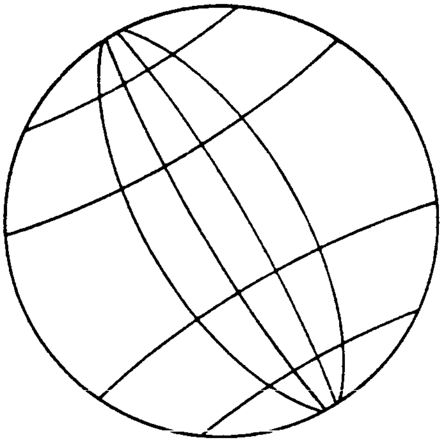
$$z = x \cos \alpha + y \sin \alpha$$

$$p = 20$$

$$a = \pi/6$$

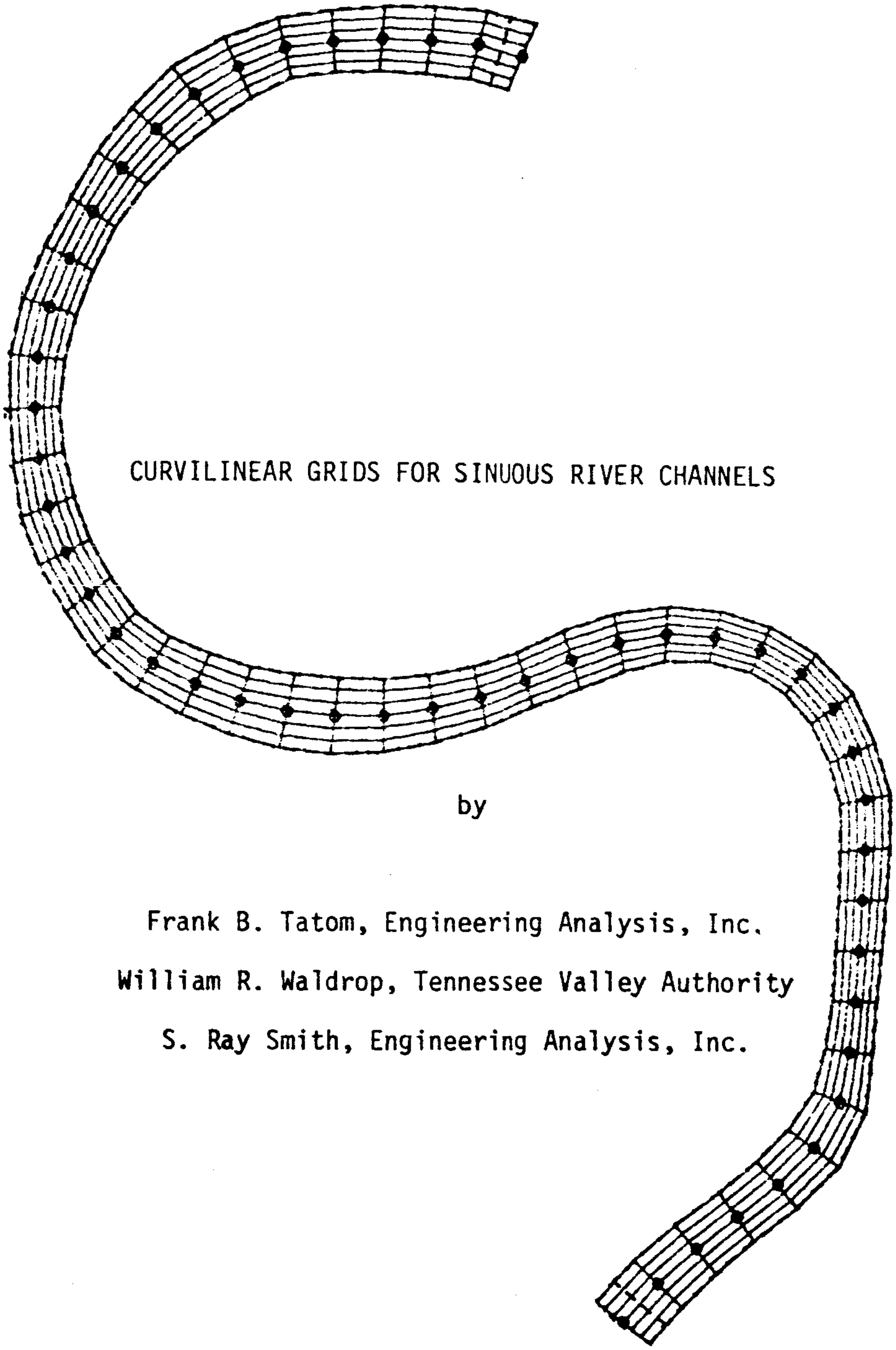
SA-6469-8

FIGURE 1 PROBLEM DOMAIN



SA-6469-9

FIGURE 2 SOLUTION GRID (SKETCH)



# CURVILINEAR GRIDS FOR SINUOUS RIVER CHANNELS

by

Frank B. Tatom, Engineering Analysis, Inc.

William R. Waldrop, Tennessee Valley Authority

S. Ray Smith, Engineering Analysis, Inc.

# CENTERLINE INTRODUCTION

In order to effectively analyze the flow in sinuous river channels a curvilinear grid system must be developed for use in the appropriate hydrodynamic code. The CENTERLINE program has been designed to generate a two-dimensional grid for this purpose.

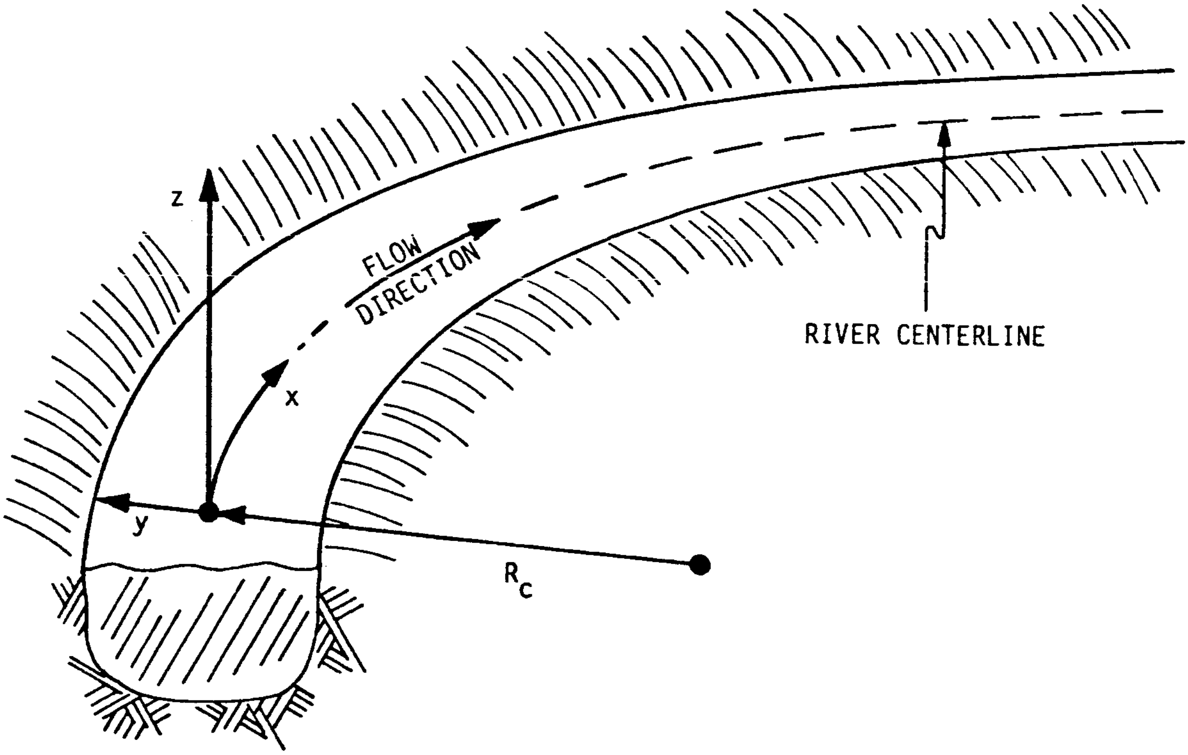
The Cartesian coordinates of a series of points along the boundaries of the sinuous channel represent the primary input to CENTERLINE. The program calculates the location of the river centerline, the distance downstream along the centerline, and both radius of curvature and channel width, as a function of such distance downstream. These parameters form the basis for the generation of the curvilinear grid.

Based on input values for longitudinal and lateral grid spacing, the corresponding grid system is generated and a file is created containing the appropriate parameters for use in the associated explicit finite difference hydrodynamic programs. Because of the option for a nonuniform grid, grid spacing can be concentrated in areas containing the largest flow gradients.

For the case of sinuous channels of constant or nearly constant width the resulting curvilinear grid is orthogonal. The grid generation procedure also provides for dividing the overall flow area under consideration into a series of regions connected along common boundaries. This concept of multiple regions tends to improve computational efficiency.

For many sinuous channels the assumption of constant width is not appropriate. In such situations CENTERLINE generates a nonorthogonal grid which takes into account the nonuniform channel width.

The CENTERLINE program is currently operational and has been used successfully in conjunction with both two- and three-dimensional incompressible hydrodynamic programs. To the authors' knowledge, it is the only *curvilinear* grid program currently coupled with *operational* incompressible hydrodynamic programs for computing two- and three-dimensional river flows.



Basic Curvilinear Coordinate System

## GOVERNING EQUATIONS FOR INCOMPRESSIBLE CURVILINEAR FLOW

CONTINUITY:

$$\frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z u) + \frac{\partial}{\partial y}(h_z h_x v) + \frac{\partial}{\partial z}(h_x h_y w)\right] = 0$$

X-MOMENTUM:

$$\begin{aligned}
&\rho\Big[\frac{\partial u}{\partial t} + \frac{u}{h_x}\frac{\partial u}{\partial x} + \frac{v}{h_y}\frac{\partial u}{\partial y} + \frac{w}{h_z}\frac{\partial u}{\partial z} \\
&\qquad - v\left(\frac{v}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{u}{h_x h_y}\frac{\partial h_x}{\partial y}\right) + w\left(\frac{u}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{w}{h_z h_x}\frac{\partial h_z}{\partial x}\right)\Big] \\
&\quad = \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xx}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yx}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zx})\right] \\
&\qquad + \frac{\sigma_{xy}}{h_x h_y}\frac{\partial h_x}{\partial y} + \frac{\sigma_{zx}}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{\sigma_{yy}}{h_x h_y}\frac{\partial h_y}{\partial x} - \frac{\sigma_{zz}}{h_x h_z}\frac{\partial h_z}{\partial x} + F_x
\end{aligned}$$

Y-MOMENTUM:

$$\begin{aligned}
&\rho\Big[\frac{\partial v}{\partial t} + \frac{u}{h_x}\frac{\partial v}{\partial x} + \frac{v}{h_y}\frac{\partial v}{\partial y} + \frac{w}{h_z}\frac{\partial v}{\partial z} \\
&\qquad - w\left(\frac{w}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{v}{h_y h_z}\frac{\partial h_y}{\partial z}\right) + u\left(\frac{v}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{u}{h_x h_y}\frac{\partial h_x}{\partial y}\right)\Big] \\
&\quad = \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xy}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yy}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zy})\right] \\
&\qquad + \frac{\sigma_{yz}}{h_y h_z}\frac{\partial h_y}{\partial z} + \frac{\sigma_{xy}}{h_y h_x}\frac{\partial h_y}{\partial x} - \frac{\sigma_{zz}}{h_y h_z}\frac{\partial h_z}{\partial y} - \frac{\sigma_{xx}}{h_y h_x}\frac{\partial h_x}{\partial y} + F_y
\end{aligned}$$

Z-MOMENTUM :

$$
\begin{aligned}
&\rho\left[\frac{\partial w}{\partial t} + \frac{u}{h_x}\frac{\partial w}{\partial x} + \frac{v}{h_y}\frac{\partial w}{\partial y} + \frac{w}{h_z}\frac{\partial w}{\partial z}\right. \\
&\qquad \left. - u\left(\frac{u}{h_x h_z}\frac{\partial h_x}{\partial z} - \frac{w}{h_z h_x}\frac{\partial h_z}{\partial x}\right) + v\left(\frac{w}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{v}{h_y h_z}\frac{\partial h_y}{\partial z}\right)\right] \\
&\quad = \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}(h_y h_z \sigma_{xz}) + \frac{\partial}{\partial y}(h_z h_x \sigma_{yz}) + \frac{\partial}{\partial z}(h_x h_y \sigma_{zz})\right] \\
&\qquad + \frac{\sigma_{zx}}{h_x h_z}\frac{\partial h_z}{\partial x} + \frac{\sigma_{yz}}{h_z h_y}\frac{\partial h_z}{\partial y} - \frac{\sigma_{xx}}{h_z h_x}\frac{\partial h_x}{\partial z} - \frac{\sigma_{yy}}{h_z h_y}\frac{\partial h_y}{\partial z} + F_z
\end{aligned}
$$

ENERGY :

$$
\begin{aligned}
&\frac{\partial T}{\partial t} + \frac{u}{h_x}\frac{\partial T}{\partial x} + \frac{v}{h_y}\frac{\partial T}{\partial y} + \frac{w}{h_z}\frac{\partial T}{\partial z} \\
&\quad = \frac{1}{h_x h_y h_z}\left[\frac{\partial}{\partial x}\left(\frac{\alpha_x h_y h_z}{h_x}\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{\alpha_y h_z h_x}{h_y}\frac{\partial T}{\partial y}\right)\right. \\
&\qquad \left. + \frac{\partial}{\partial z}\left(\frac{\alpha_z h_x h_y}{h_z}\frac{\partial T}{\partial z}\right)\right]
\end{aligned}
$$

# COMPUTATION OF METRIC COEFFICIENTS

FUNDAMENTAL CONSIDERATIONS:

- APPEAR IN GOVERNING EQUATIONS
- ONLY $h_x$ REQUIRES COMPUTATION
- EVALUATED FOR EACH GRID POINT
- DERIVATIVES ALSO REQUIRED

BASIC RELATIONS:

$$h_x = \frac{R_c + y}{R_c}$$

$$h_y = 1$$

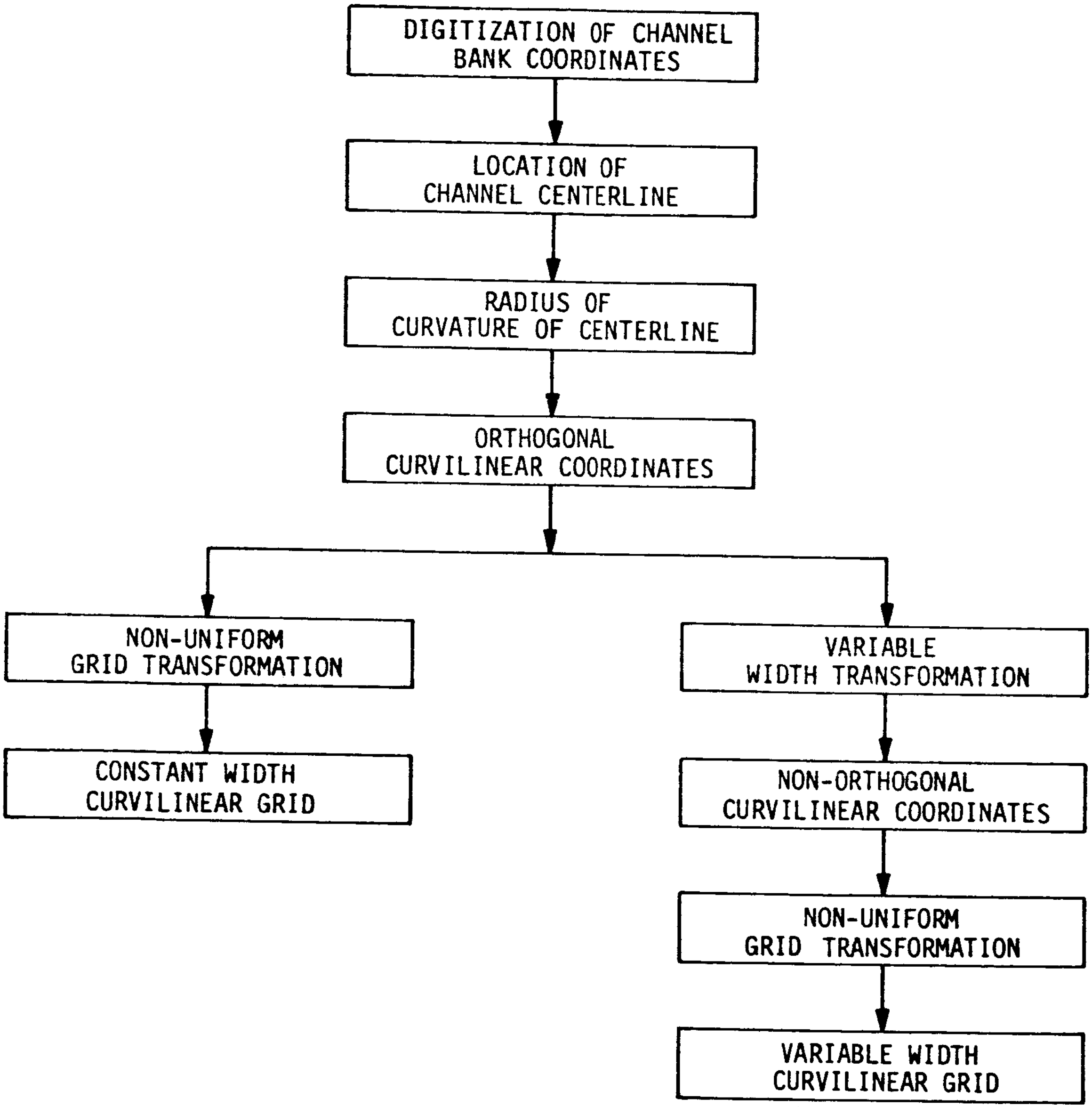
$$h_z = 1$$

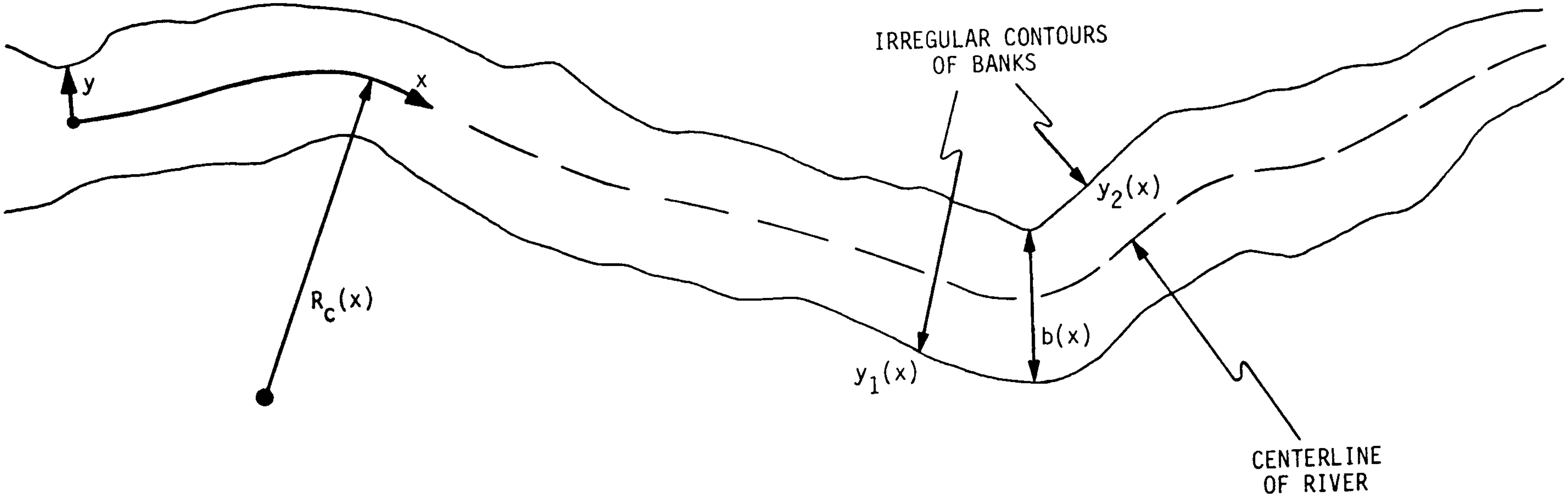
DERIVATIVES:

$$\frac{\partial h_x}{\partial x} = - \frac{y}{R_c{}^2} \frac{dR_c}{dx}$$

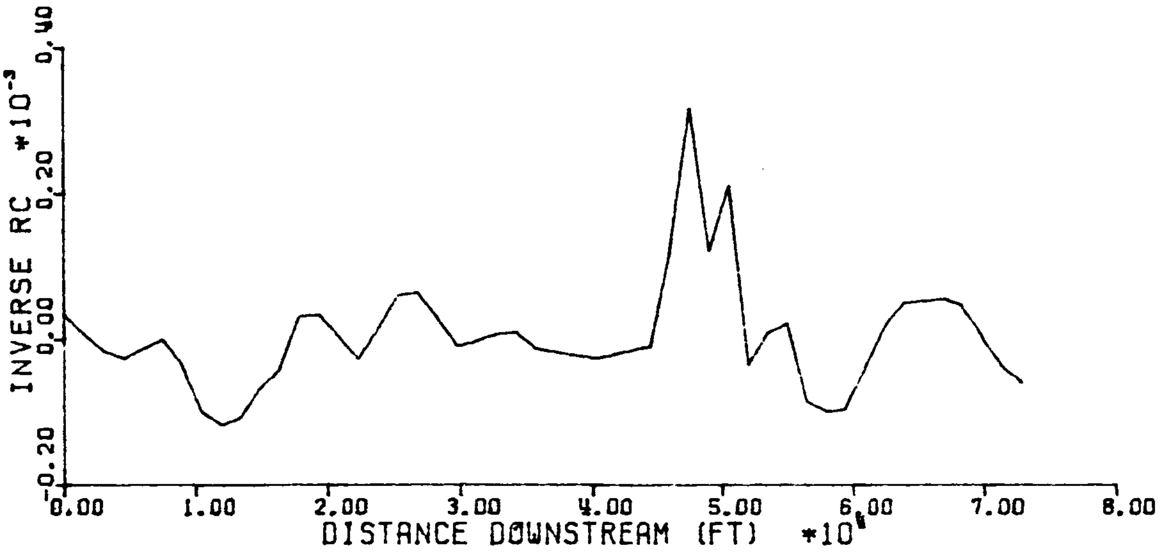
$$\frac{\partial h_x}{\partial y} = \frac{1}{R_c}$$

GENERATION OF CURVILINEAR GRIDS

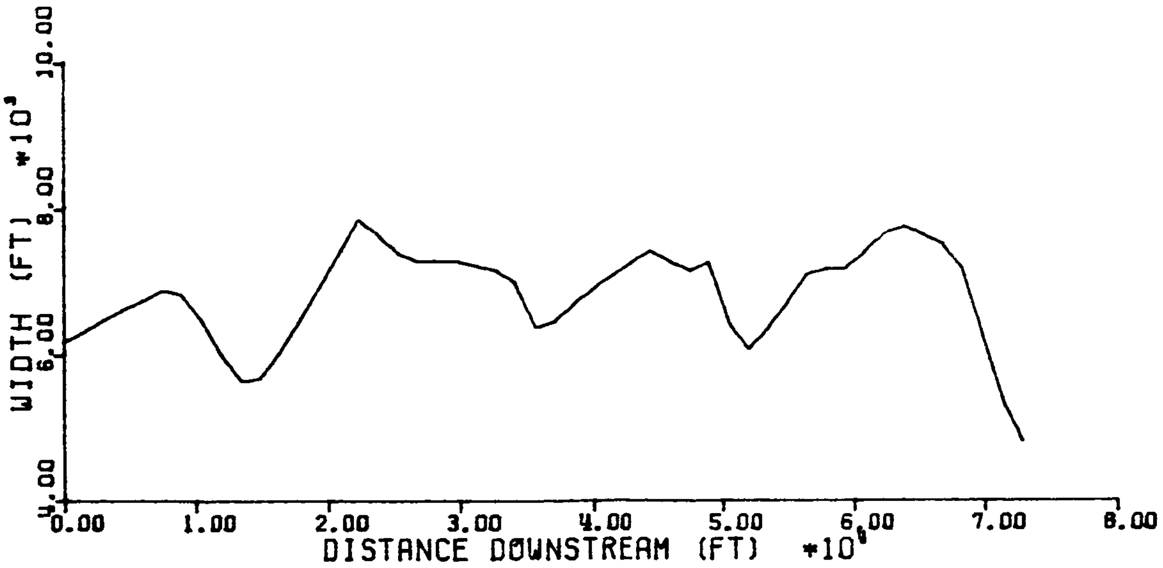

DIGITIZATION OF CHANNEL BANK COORDINATES
LOCATION OF CHANNEL CENTERLINE
RADIUS OF CURVATURE OF CENTERLINE
ORTHOGONAL CURVILINEAR COORDINATES
NON-UNIFORM GRID TRANSFORMATION
CONSTANT WIDTH CURVILINEAR GRID
VARIABLE WIDTH TRANSFORMATION
NON-ORTHOGONAL CURVILINEAR COORDINATES
NON-UNIFORM GRID TRANSFORMATION
VARIABLE WIDTH CURVILINEAR GRID



Typical Reservoir



Inverse Radius of Curvature Vs Distance Dowstream for Typical Reservoir



Channel Width Vs Distance Downstream for Typical Reservoir

COMPUTATION OF RADIUS OF CURVATURE AND CHANNEL WIDTH

- DIGITIZE CARTESIAN COORDINATES OF CHANNEL BANKS
- LOCATE GEOMETRIC CENTERLINE
- COMPUTE DISTANCE ALONG CENTERLINE, $x$
- COMPUTE RADIUS OF CURVATURE, $R_c(x)$
- COMPUTE CHANNEL WIDTH, $b(x)$

# VARIABLE WIDTH TRANSFORMATION

BASIC TRANSFORMATION:

$$b = y_2(x) - y_1(x)$$

$$X = x$$

$$Y = y/b$$

TRANSFORMATION DERIVATIVES:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial X} + \frac{\partial f}{\partial Y} Y'$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial Y} \frac{1}{b}$$

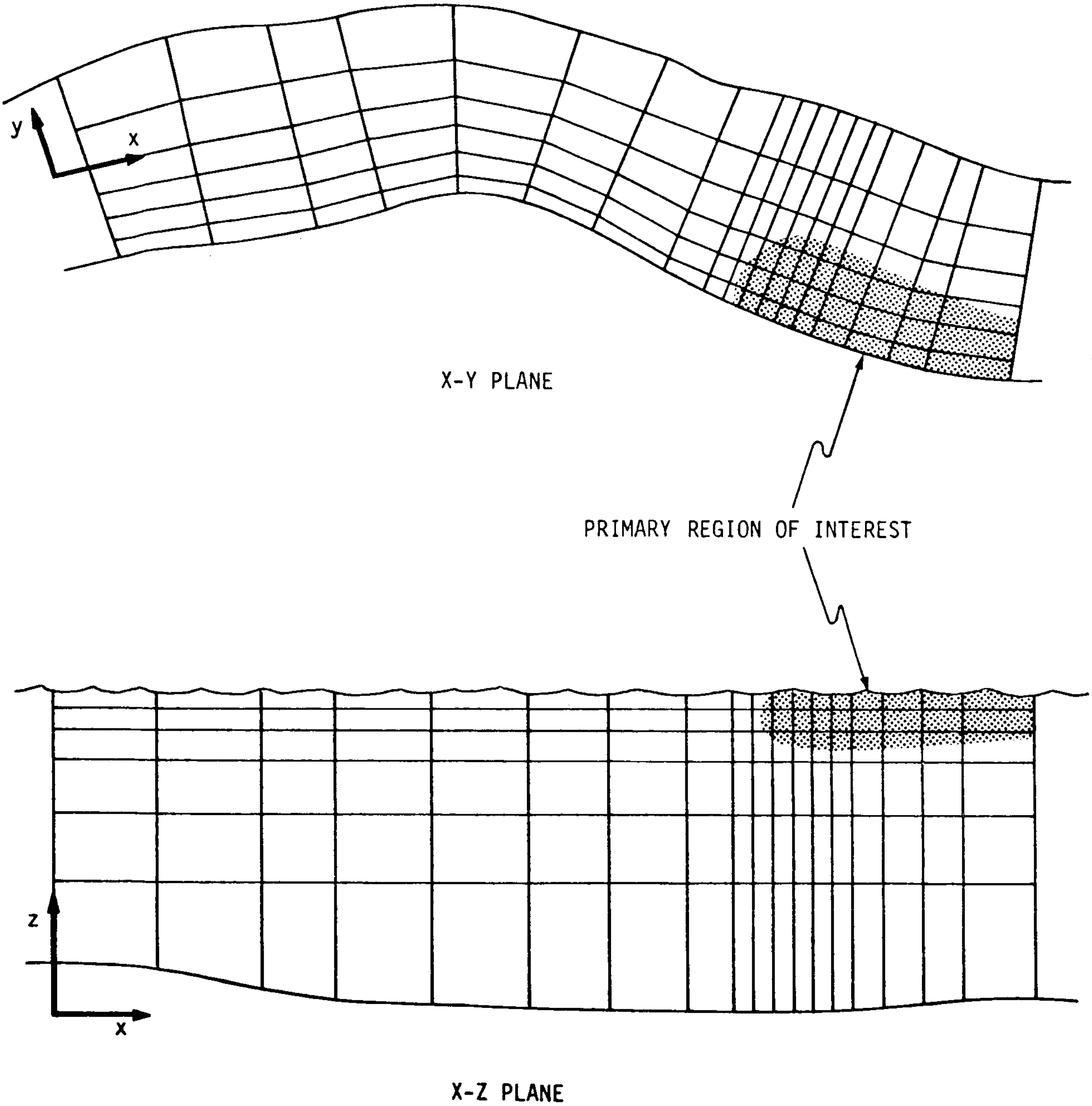
$$\frac{\partial^2 f}{\partial x^2} = \frac{\partial^2 f}{\partial X^2} + 2 \frac{\partial^2 f}{\partial X \partial Y} Y' + \frac{\partial^2 f}{\partial Y^2} (Y')^2 + \frac{\partial f}{\partial Y} Y''$$

$$\frac{\partial^2 f}{\partial y^2} = \frac{\partial^2 f}{\partial Y^2} \frac{1}{b^2}$$

where

$$Y' = - \frac{Y}{b} \frac{db}{dx}$$

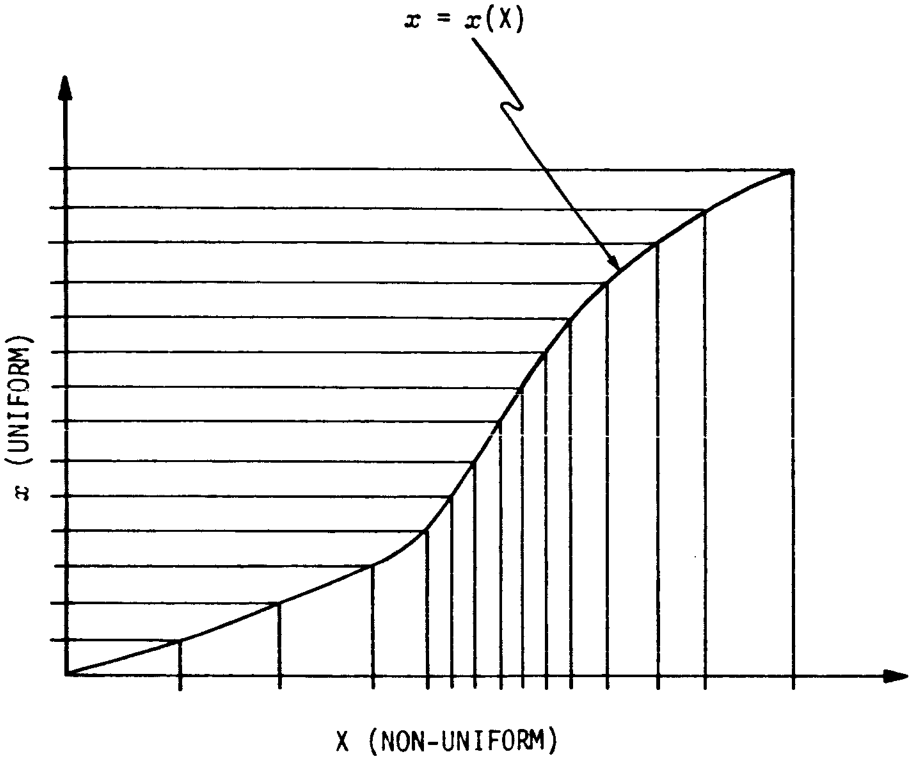
$$Y'' = \frac{2Y}{b^2} \frac{db}{dx} - \frac{Y}{b} \frac{d^2 b}{dx^2}$$



X-Y PLANE

X-Z PLANE

Non-Uniform Grid System



Relationship Between Non-Uniform and Uniform Grids

## TRANSFORMATION FROM NON-UNIFORM TO UNIFORM GRID

PROCEDURE:

- IDENTIFY "REGIONS OF INTEREST"
- INPUT DESIRED GRID SPACING
- GENERATE TRANSFORMATION DERIVATIVES

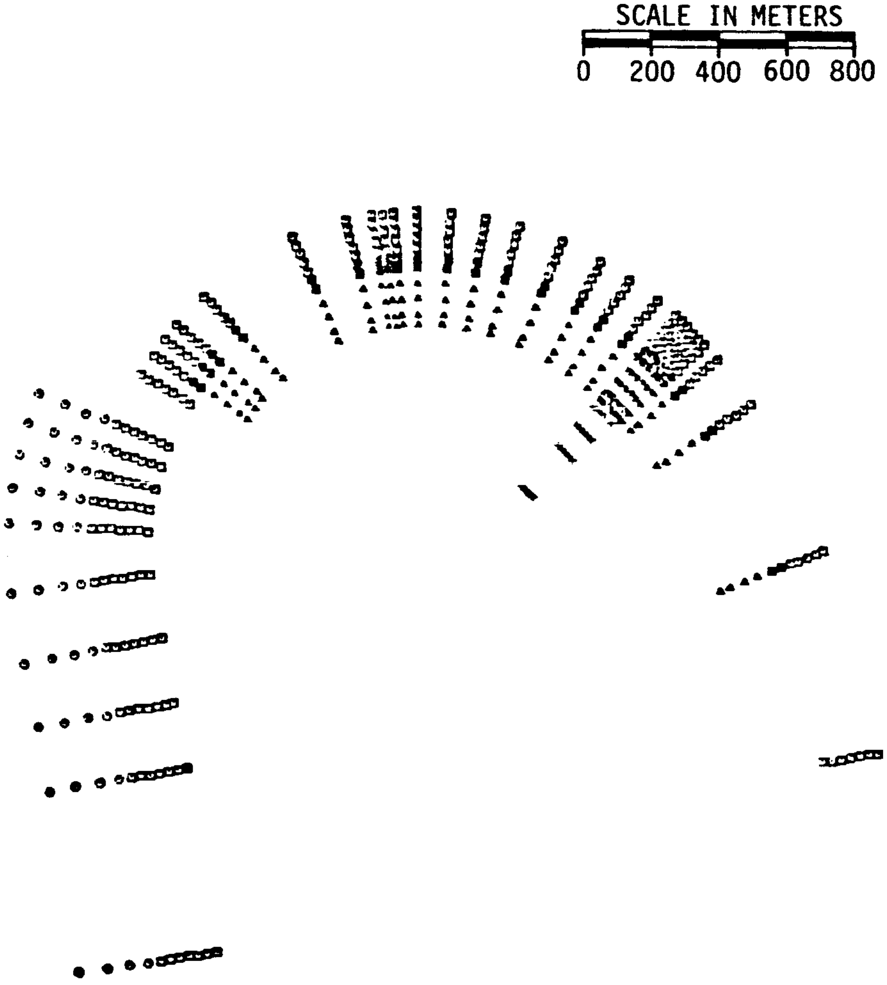
BASIC TRANSFORMATION:

$$\left.\begin{aligned} x &= x(\mathrm{X}) \\ y &= y(\mathrm{Y}) \\ z &= z(\mathrm{Z}) \end{aligned}\right\} \quad \begin{array}{l}\text{ANALYTICAL TRANSFORMATION} \\ \text{FUNCTIONS NOT REQUIRED}\end{array}$$

TRANSFORMATION DERIVATIVES:

$$\frac{\partial g}{\partial \mathrm{X}} = \frac{\partial g}{\partial x} \frac{\partial x}{\partial \mathrm{X}}$$

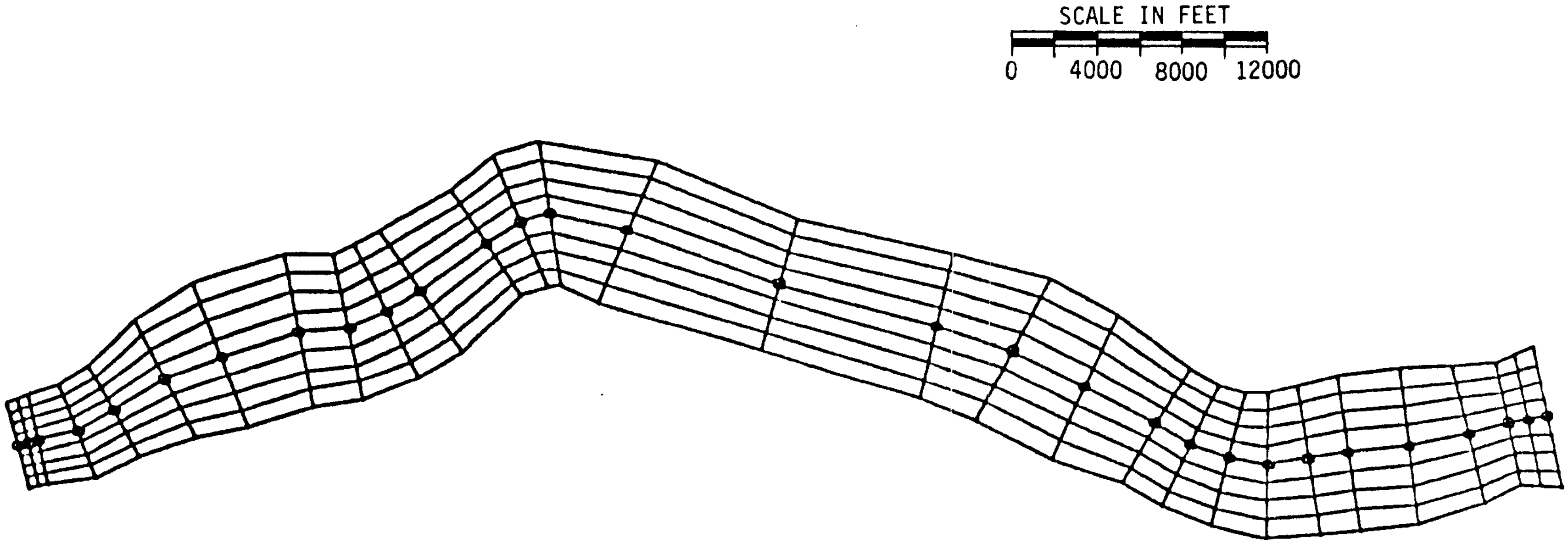
$$\frac{\partial^2 g}{\partial \mathrm{X}^2} = \frac{\partial g}{\partial x} \frac{\partial^2 x}{\partial \mathrm{X}^2} + \frac{\partial^2 g}{\partial x^2} \left(\frac{\partial x}{\partial \mathrm{X}}\right)^2$$



Curvilinear Grid for Cumberland River Segment

CURVILINEAR GRID FOR CUMBERLAND RIVER SEGMENT

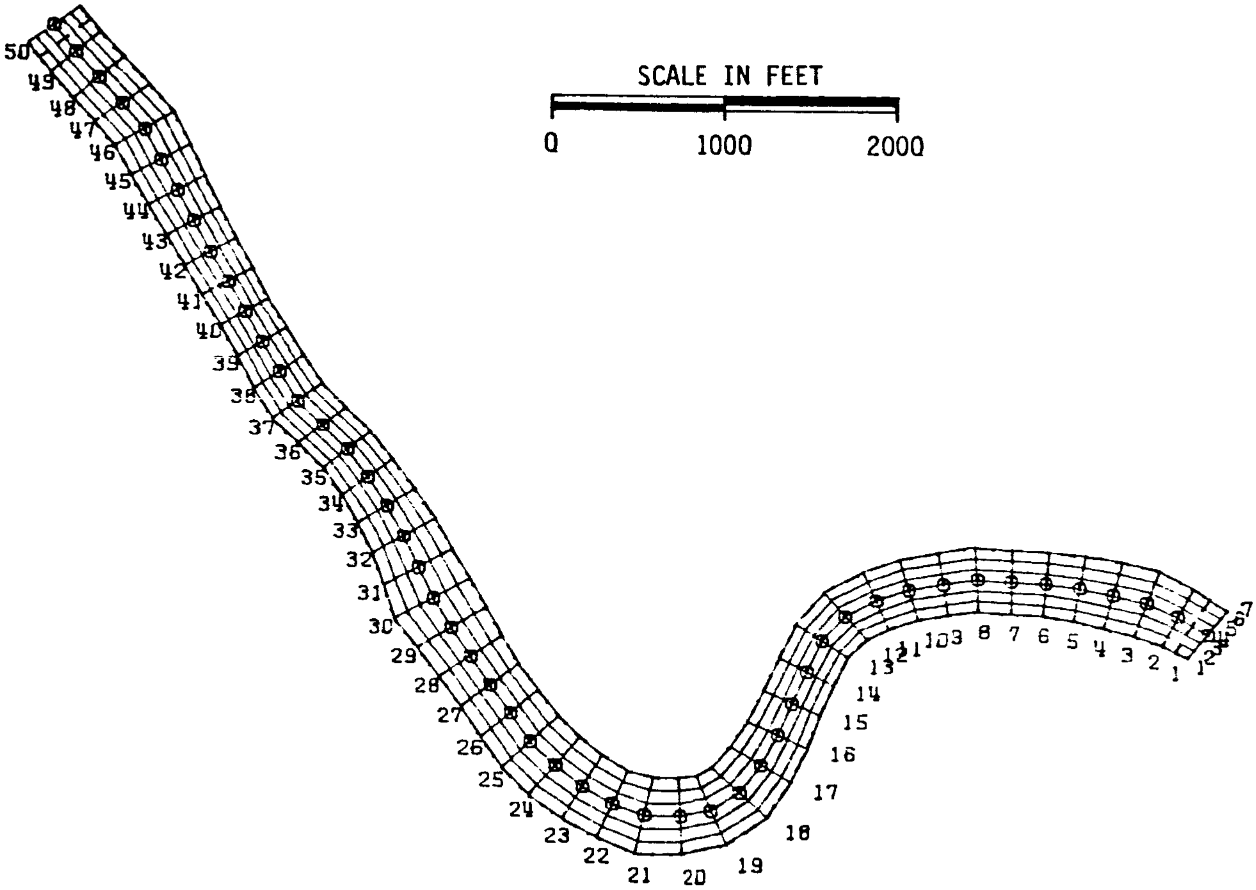
- NEAR TVA GALATIN STEAM PLANT
- CONSTANT WIDTH CHANNEL
- NON-UNIFORM GRID (x, y, & z)
- 4 CONNECTED REGIONS
- USED IN 3-D FLOW COMPUTATIONS



Curvilinear Grid for Tennessee River, Wilson Reservoir

CURVILINEAR GRID FOR TENNESSEE RIVER, WILSON RESERVOIR

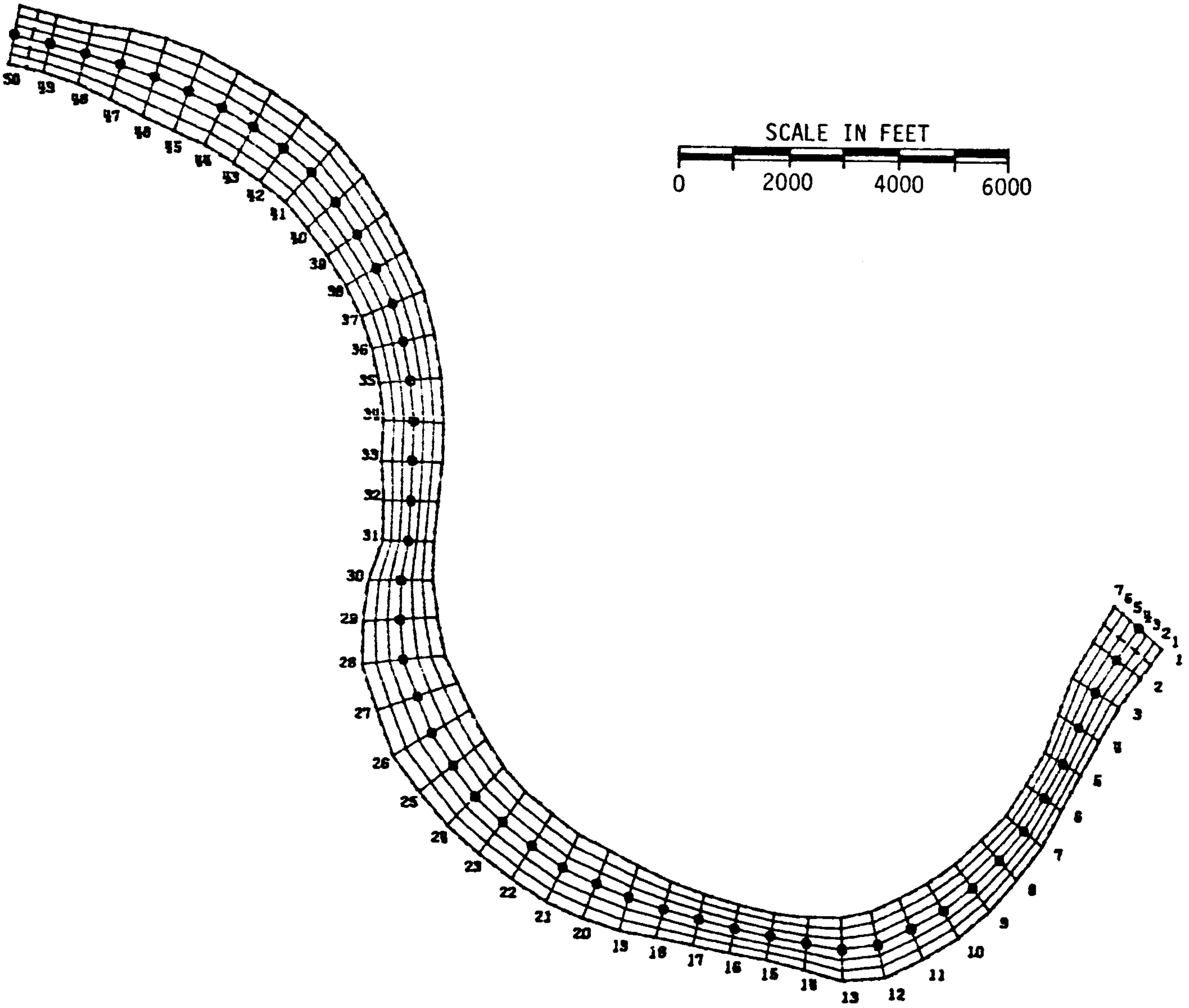
- BETWEEN WHEELER AND WILSON DAMS
- VARIABLE WIDTH CHANNEL
- NON-UNIFORM GRID (x only)
- USED IN 2-D DEPTH-AVERAGED FLOW COMPUTATION



Curvilinear Grid for Green River Segment

## CURVILINEAR GRID FOR GREEN RIVER SEGMENT

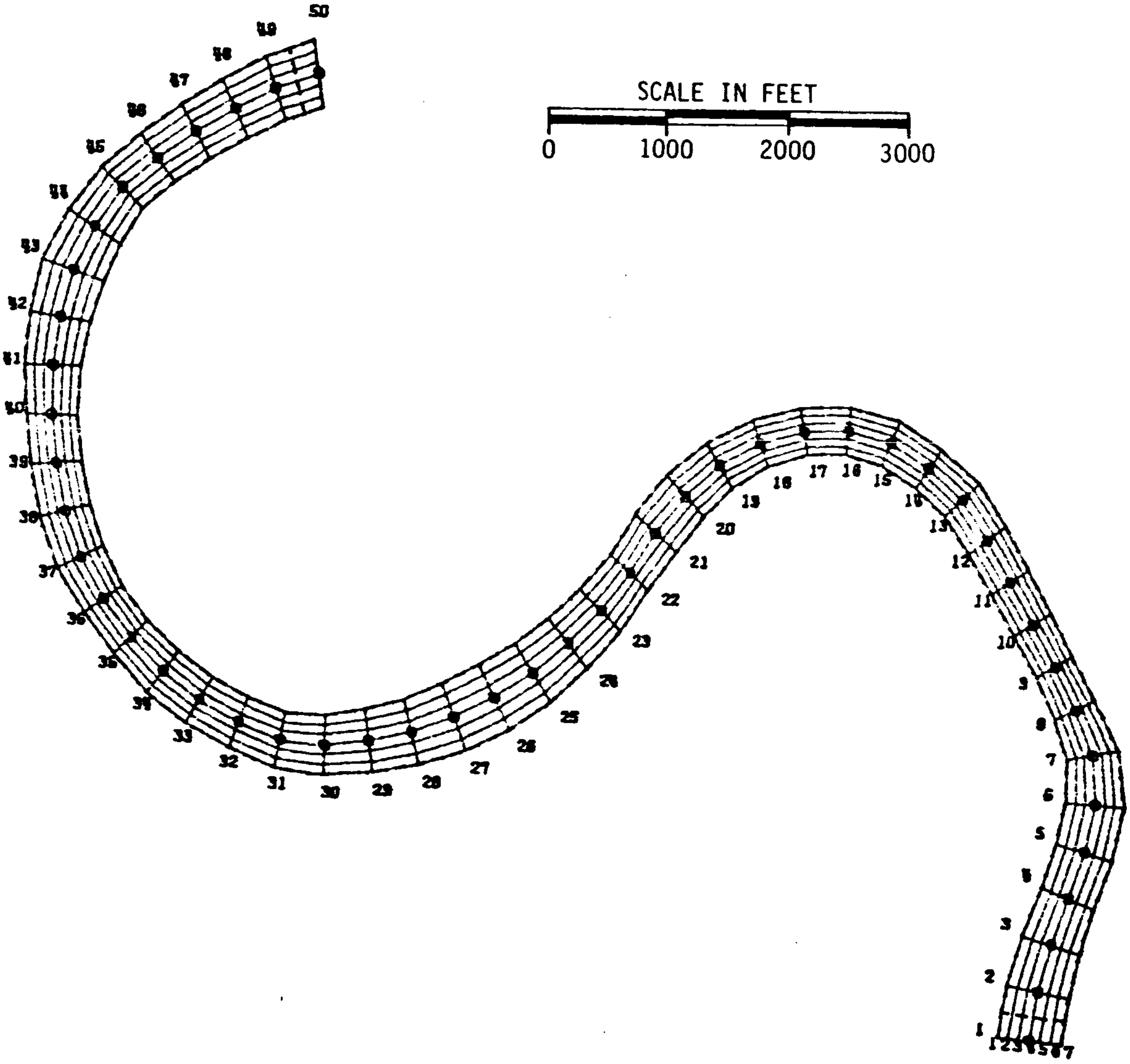
- NEAR PARADISE STEAM PLANT
- MODERATE SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID



Curvilinear Grid for Tennessee River, Wheeler Reservoir

CURVILINEAR GRID FOR TENNESSEE RIVER, WHEELER RESERVOIR

- NEAR REDSTONE ARSENAL
- MODERATE SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID

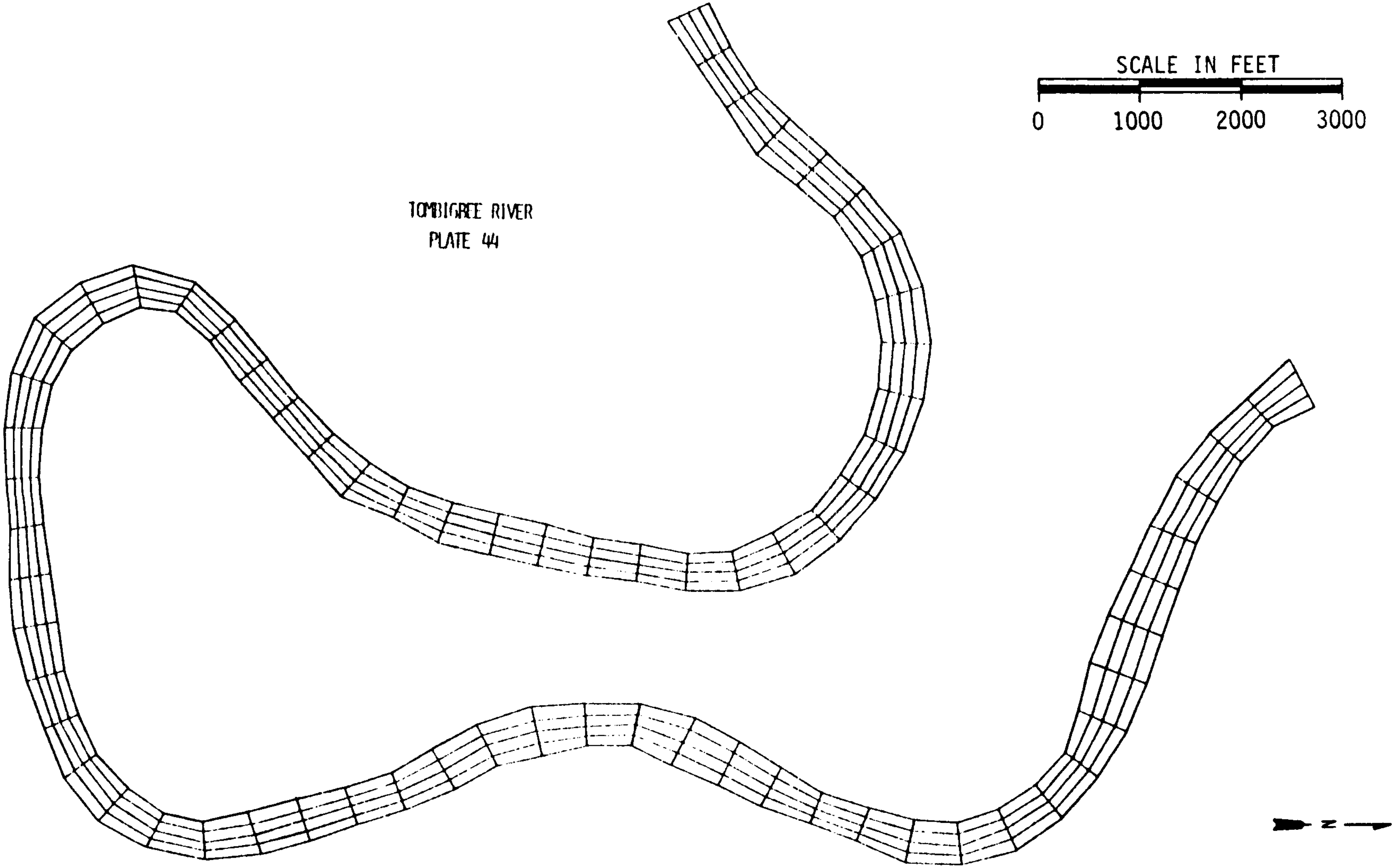


Curvilinear Grid for Little Tennessee River Segment

CURVILINEAR GRID FOR LITTLE TENNESSEE RIVER SEGMENT

- PART OF TELLICO LAKE
- HIGH SINUIOSITY
- VARIABLE WIDTH
- UNIFORM GRID

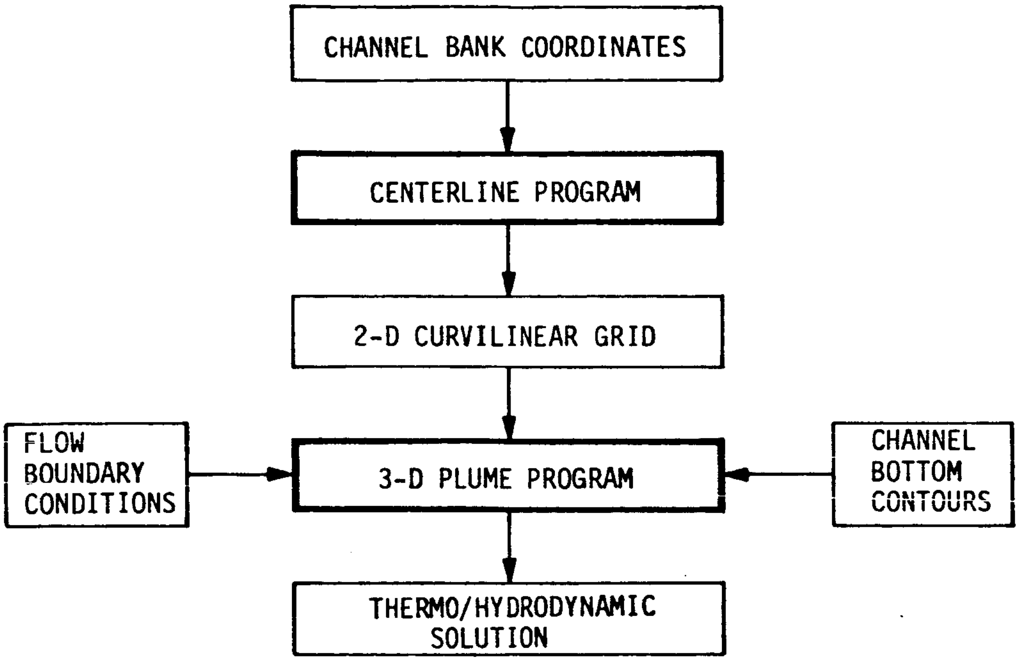
TOMBIGREE RIVER
PLATE 44



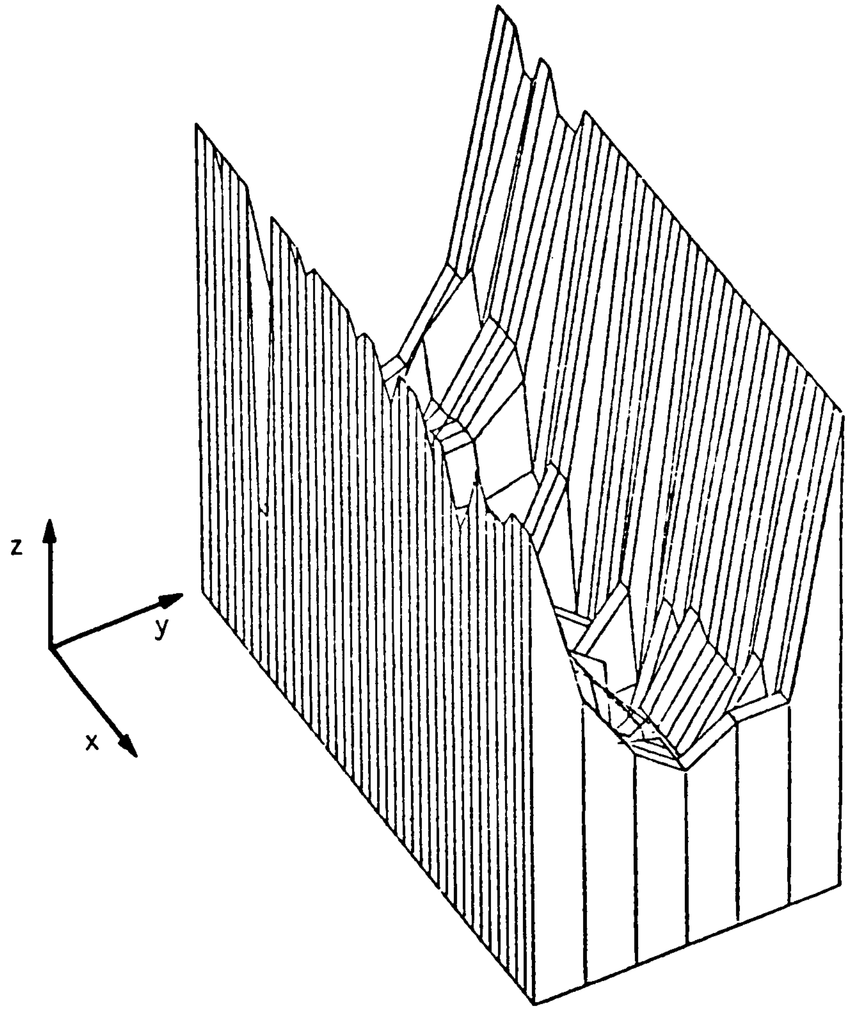
Curvilinear Grid for Tombigbee River Segment

CURVILINEAR GRID FOR TOMBIGBEE RIVER SEGMENT

- PORTION OF TENNESSEE - TOMBIGBEE WATERWAY
- EXTREME SINUOSITY
- VARIABLE WIDTH
- UNIFORM GRID



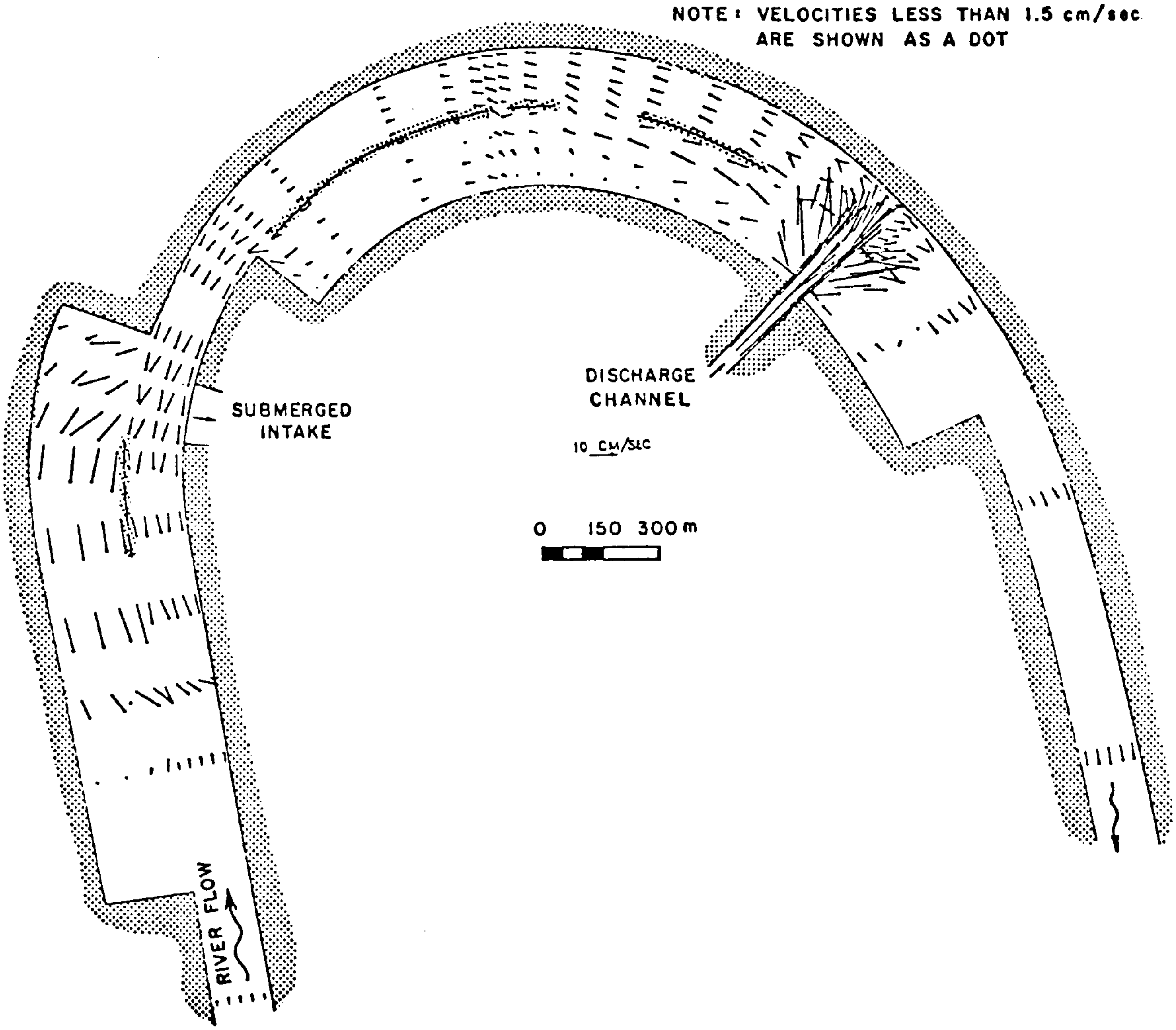
Integration of CENTERLINE Program with 3-D PLUME Program



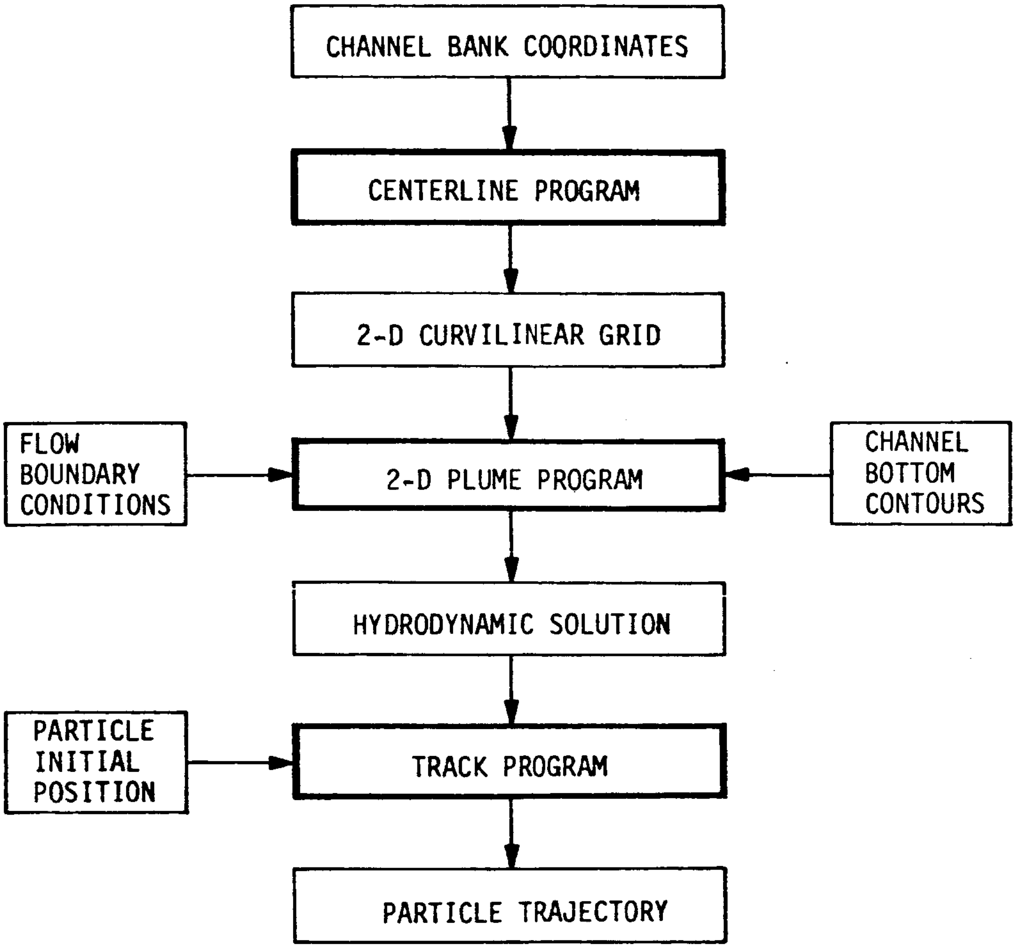
Non-Uniform Bottom of Cumberland River Segment

## NON-UNIFORM BOTTOM CONSIDERATIONS

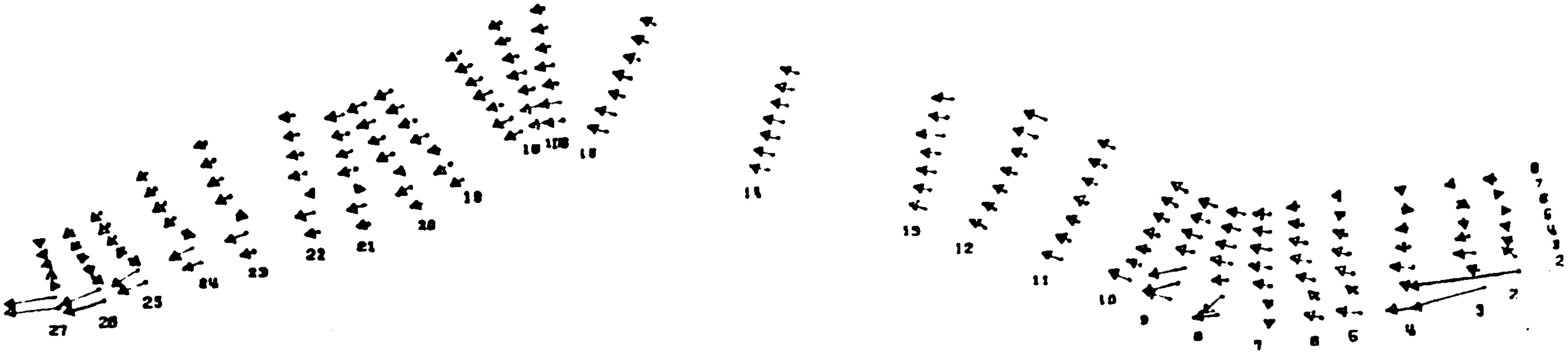
- BOTTOM PROFILES BASED ON SOUNDINGS
- LONGITUDINAL AND TRANSVERSE VARIATIONS ACCEPTED
- GRID SPACING LIMITS RESOLUTION OF BOTTOM SHAPE
- BOTTOM PROFILES NOT USED FOR TRANSFORMATION



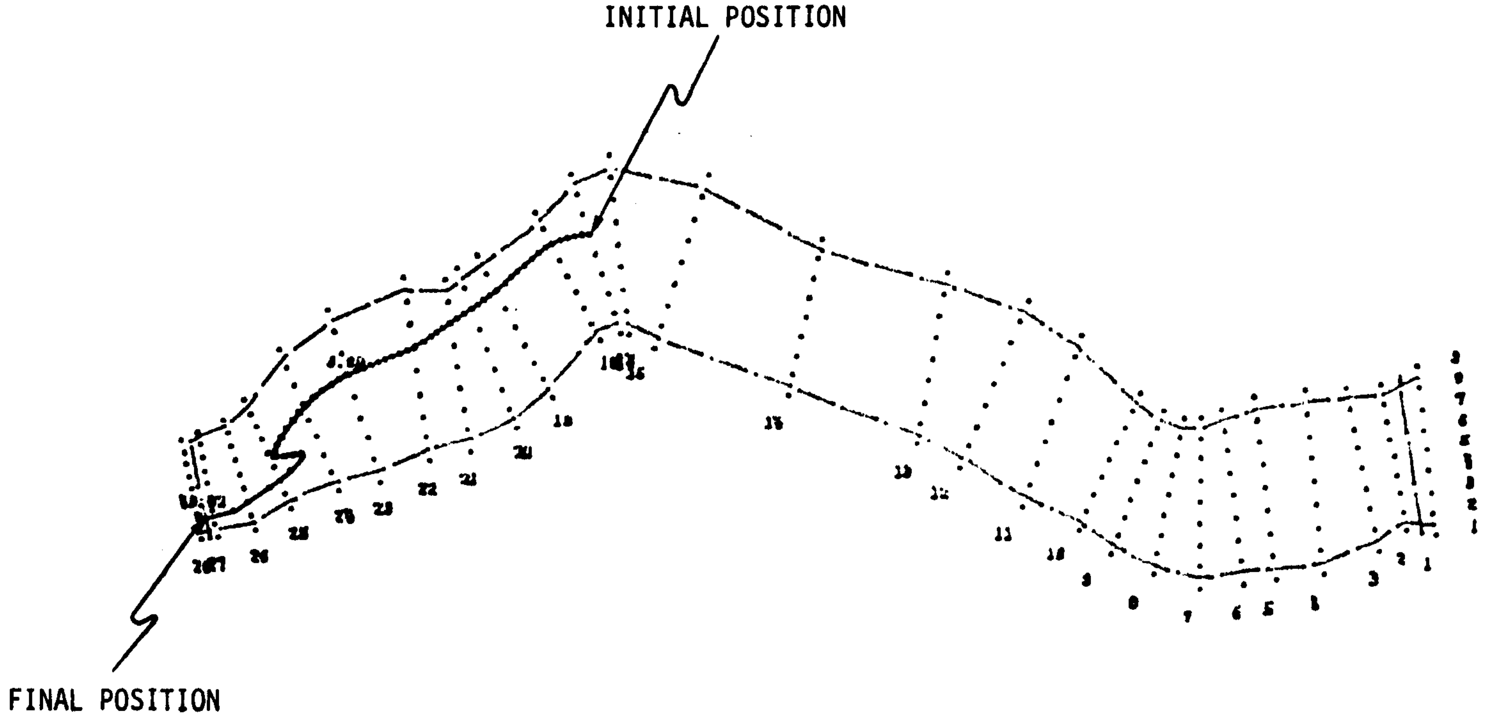
Velocity Vector Plot for Cumberland River Segment



Integration of CENTERLINE Program with 2-D PLUME and TRACK Programs



Velocity Vector Plot for the Tennessee River, Wilson Reservoir



Particle Trajectory Plot from TRACK in Tennessee River, Wilson Reservoir

## CENTERLINE SUMMARY

- APPLICABLE TO SINUOUS RIVER CHANNELS
- CURRENTLY OPERATIONAL
- DIGITIZATION OF CHANNEL COORDINATES
- CONSTANT/VARIABLE CHANNEL WIDTH OPTIONS
- UNIFORM/NON-UNIFORM GRID OPTIONS
- PRESENTLY COUPLED WITH 2-D AND 3-D HYDRODYNAMIC MODELS